

LAPORAN RESMI PRAKTIKUM 6

REKURSIF



Dibuat Oleh :

Nama : Ummiyatun
NIM : 21091397039
Kelas : A

Dosen Pengampu :

Andi Iwan Nurhidayat, S.KOM., M.T.

UNIVERSITAS NEGERI SURABAYA
PROGRAM VOKASI
PRODI D4 MANAJEMEN INFORMATIKA
TAHUN 2021

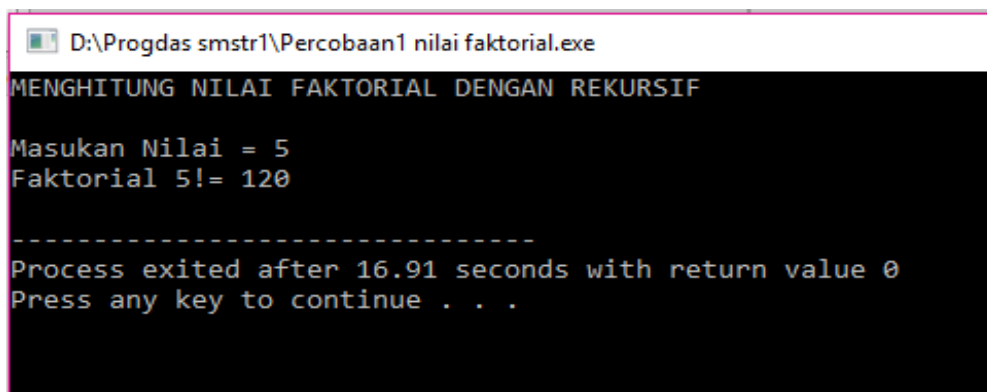
D. PERCOBAAN

- Percobaan 1 : Fungsi rekursif untuk menghitung nilai faktorial

Input

```
#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));
}
```

Output



```
D:\Progdas smstr1\Percobaan1 nilai faktorial.exe
MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF

Masukan Nilai = 5
Faktorial 5!= 120

-----
Process exited after 16.91 seconds with return value 0
Press any key to continue . . .
```

- Percobaan 2 : Fungsi rekursi untuk menampilkan deret fibonanci

Input

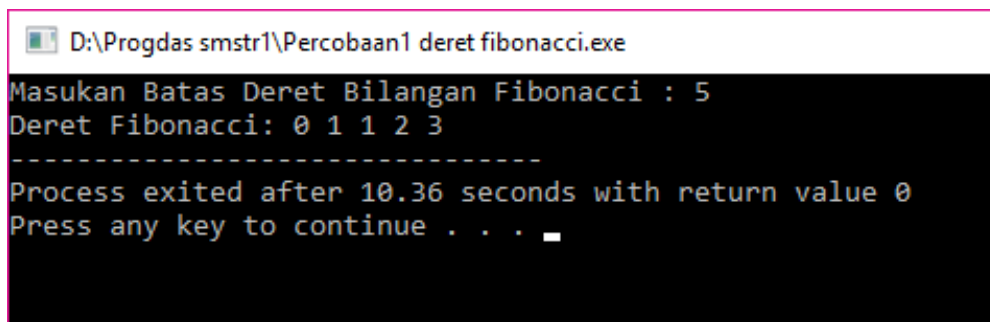
```
#include <iostream>
using namespace std;
int fibonacci(int m) {
    if (m == 0 || m ==1)
    {
        return m;
    }
    else
    {
```

```

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;
}

```

Output



```

D:\Progdas smstr1\Percobaan1 deret fibonacci.exe
Masukan Batas Deret Bilangan Fibonacci : 5
Deret Fibonacci: 0 1 1 2 3
-----
Process exited after 10.36 seconds with return value 0
Press any key to continue . . .

```

- **Percobaan 3 : Fungsi rekursi untuk menentukan bilangan prima atau bukan prima**

Input

```

#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;

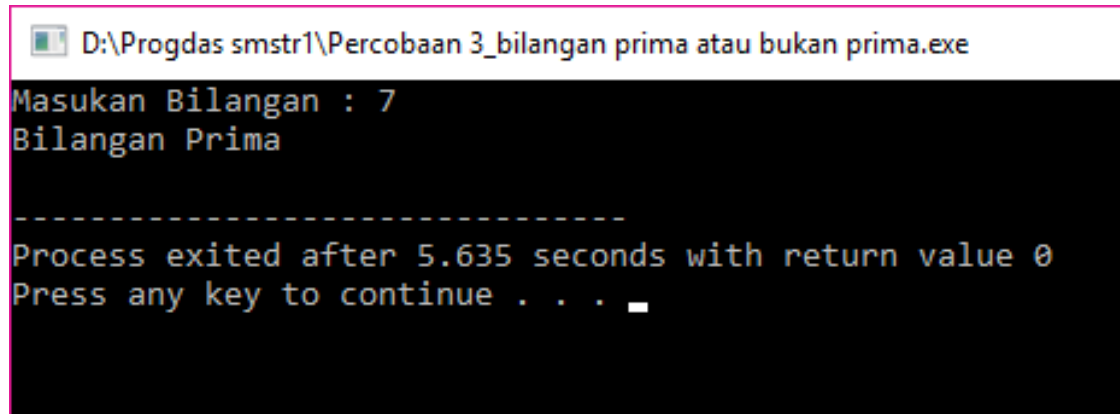
```

```

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

```

Output



```

D:\Progdas smstr1\Percobaan 3_bilangan prima atau bukan prima.exe
Masukan Bilangan : 7
Bilangan Prima

-----
Process exited after 5.635 seconds with return value 0
Press any key to continue . . .

```

▪ Percobaan 4 : Fungsi rekursi untuk menghitung pangkat

Input

```

#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); }

```

Ouput

```
D:\Progdas smstr1\Percobaan 1_menghitung pangkat.exe
FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT

Masukan Nilai X = 4
Masukan Nilai Y = 2

4 Dipangkatkan 2 = 16

-----
Process exited after 30.27 seconds with return value 0
Press any key to continue . . .
```

E. LATIHAN

1. Sorce Code

```
#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<<"\tProgram rekursif menghitung segitiga Pascal \n";
    cout<<"\t===== \n";
    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;
}
```

Analisa : Dari program tersebut dapat dilihat bahwa fungsi untuk segitiga pascal memanggil fungsinya sendiri untuk menampilkan hasil dari penjumlahan pascal.

2. Sorce Code

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int compare (const void * a, const void * b);
void allLexicographicRecur (char *str, char* data, int last, int index)
{
    int i, len = strlen(str);
    for ( i=0; i<len; i++ )
    {
        data[index] = str[i] ;
        if (index == last)
            printf("%s ", data);
        else
            allLexicographicRecur (str, data, last, index+1);
    }
}
void allLexicographic(char *str)
{
    int len = strlen (str) ;
    char *data = (char *) malloc (sizeof(char) * (len + 1)) ;
    data[len] = '\0';
    qsort(str, len, sizeof(char), compare);
    allLexicographicRecur (str, data, len-1, 0);
    free(data);
}
int compare (const void * a, const void * b)
{
    return ( *(char *)a - *(char *)b );
}
int main()
{
    char str[100];
    printf("Masukan jumlah karakter: ");gets(str);
    printf("Karakternyanya adalah: \n", str);
    allLexicographic(str);
    return 0;
}
```

Analisa : program di atas merupakan program rekursif untuk mencetak permutasi lexicograpich dengan adanya repitisi.

3. Source Code

```
#include<stdio.h>
#include<stdlib.h>
#define size 10
int binsearch(int[], int, int, int);
int binsearch(int a[], int x, int low, int high) {
    int mid;
    if (low > high)
        return -1;
    mid = (low + high) / 2;
    if (x == a[mid]) {
```

```

    return (mid);
} else if (x < a[mid]) {
    binsearch(a, x, low, mid - 1);
} else {
    binsearch(a, x, mid + 1, high);
}
}
int main() {
    int num, i, key, position;
    int low, high, list[size];
    printf("\t====Program Binary Search====\n");
    printf("\nMasukkan Banyak data: ");
    scanf("%d", &num);
    for (i = 0; i < num; i++) {
        printf("Masukkan data ke %d : ", i);
        scanf("%d", &list[i]);
    }
    low = 0;
    high = num - 1;
    printf("\nMasukan data yang ingin di cari : ");
    scanf("%d", &key);
    position = binsearch(list, key, low, high);
    if (position != -1) {
        printf("\nData %d berada pada Indeks ke - %d", key, (position));
    } else
        printf("\nData tidak di temukan");
    return (0);
}

```

Analisa : Program tersebut merupakan penerapan dari pencarian Binary, berfungsi untuk mencari data dalam sebuah array dan menentukan posisi data tersebut.

4. Source Code

```

#include <stdio.h>

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

int main()
{
    int n;
    printf("Masukan Jumlah Disk:"); scanf("%d",&n);
    Hanoi(n, 'A', 'C', 'B');
    return 0;
}

```

Analisa : jika $n==1$, maka pindahkan disk 1 dari pasak... ke pasak...

5. Program tersebut merupakan program konversi angka dari Desimal ke Biner. Dalam program tersebut jika angka lebih besar daripada nol (0), maka angka atau bilangan tersebut dibagi dengan nilai 2 hingga selesai. Kemudian outputnya adalah angka sisa hasil bagi dari bilangan yang telah dibagi dengan nilai 2 tersebut.
6. Program tersebut merupakan program pencarian Boolean, yaitu hanya bisa diisi dengan salah satu dari 2 pilihan nilai : **true** atau **false**. Jika ukuran yang diinput lebih besar daripada 0, dan jika $(x[\text{size}-1]) = n$ maka akan menghasilkan kondisi true. Kemudian kondisi akan menampilkan salah ketika $(x, \text{size}-1, n)$.
7. Program tersebut merupakan program pencarian Biner Boolean. Jika $\text{akhir} < \text{awal}$ maka salah. Nilai di tengah $(\text{mid}) = (\text{awal} + \text{akhir}) / 2$, jika $(x[\text{mid}] == n)$ maka kondisi true. Jika $(x[\text{mid}] < n)$ maka mengembalikan pencarian $(x, \text{mid}+1, \text{end}, n)$ dan yang lain yaitu $(x, \text{start}, \text{mid}-1, n)$.
8. Metode Misteri mengambil a dan b sebagai argumen, terdapat dua set argumen yaitu (2,25) dan (3,11). Jika $(b == 0)$ maka kembali ke 0. Jika $(b \% 2 == 0)$, melakukan $\text{return mystery}(a+a, b/2)$, $\text{return mystery}(a+a, b/2) + a$.
9. Metode Misteri dengan mengambil a dan b sebagai argumen. Jika $(a == b)$ maka outputnya adalah a.
 $m1 = (a + b) / 2$, $m2 = (a + b + 1) / 2$. Maka $\text{mystery}(a, m1)$, $\text{mystery}(m2, b)$.
10. Dalam program tersebut terdapat rumus $f(n)$. jika $(n == 0)$ maka return 0, jika $(n == 1)$ maka return 1, jika $(n == 2)$ maka return 1, dan dioperasikan ke $2*f(n-2) + f(n-3)$.
11. - Terdapat variabel persegi (n) , jika $(n == 0)$ maka return 0, dan dioperasikan ke $\text{persegi}(n-1) + 2*n - 1$.
 - Terdapat variabel kubus (n) , jika $(n == 0)$ maka return 0, dan dioperasikan ke $\text{kubus}(n-1) + 3*(\text{square}(n)) - 3*n + 1$.