

LAPORAN PRAKTIKUM 09

Rekursif

Semester 2 Tahun Akademik 2022/2023



DOSEN PENGAMPU:

Entin Martiana Kusumaningtyas, S.kom, M.kom

PENYUSUN:

Arsyita Devanaya Arianto (3122500008)

PROGRAM STUDI VOKASI

D-III TEKNIK INFORMATIKA

**DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

Praktikum 9

Rekursif

PERCOBAAN

1. Fungsi rekursif untuk menghitung nilai factorial

Listing Program:

```
#include <stdio.h>
#include <stdlib.h>

int faktorial(int x){
    if(x==1){
        return x;
    }else{
        return x*faktorial(x-1);
    }
}

void main(){
    int N;
    printf("Masukkan N= ");
    scanf("%d", &N);
    printf("Hasil %d! = %d\n", N, faktorial(N));
}
```

Output:

```
Masukkan N = 8
Hasil 8! = 40320
```

Analisa:

Program tersebut adalah program untuk menghitung bilangan factorial. Dalam program tersebut terdapat fungsi faktorial(int x) digunakan untuk menghitung nilai faktorial dari bilangan x. Jika bilangan x bernilai 1 maka akan mengembalikan nilai x itu sendiri yaitu 1. Jika bilangan selain 1 akan mengembalikan nilai $x \times \text{faktorial}(x-1)$. Fungsi main() dalam program tersebut digunakan untuk menginputkan nilai. Kemudian memanggil fungsi faktorial dengan argument variabel yang diinputkan yaitu N. Lalu program akan menampilkan hasil dengan menggunakan printf().

2. Fungsi menghitung nilai faktorial dengan rekursi tail

Listing Program:

```
#include <stdio.h>
#include <stdlib.h>

int faktorial(int x, int a){
    if(x==1){
        return a;
    }else{
        return faktorial(x-1,x*a);
    }
}

void main(){
    int N;
    printf("Masukkan N= ");scanf("%d", &N);
    printf("Hasil %d! = %d\n",N,faktorial(N,1));
}
```

Output:

```
Masukkan N= 8
Hasil 8! = 40320
```

Analisa:

Program tersebut merupakan program untuk menghitung bilangan faktorial menggunakan metode rekursi tail. Dalam program tersebut terdapat fungsi faktorial(int x, int a) digunakan untuk menghitung bilangan faktorial. Parameter x didefinisikan sebagai bilangan bulat yang ingin dihitung bilangan faktorialnya dan a sebagai hasil perkalian sementara. Jika x bernilai 1 maka akan mengembalikan hasil a, sedangkan jika x tidak sama dengan 1 maka mengembalikan hasil fungsi itu sendiri dengan parameter (x-1, x*a). Fungsi main() digunakan untuk menginputkan bilangan faktorial, lalu memanggil fungsi dengan argument (bilangan yang diinputkan, 1), lalu menampilkan hasil.

3. Fungsi rekursi untuk menampilkan deret fibonanci

Listing Program:

```
#include <stdio.h>
#include <stdlib.h>

int fibo(int x){
    if(x<=0||x<=1){
        return x;
    }else{
        return fibo(x-2)+fibo(x-1);
    }
}

void main(){
    int n;
    printf("Masukkan jumlah deret= ");
    scanf("%d", &n);

    printf("Deret fibonanci dari %d = ", n);
    for(int i=0;i<n;i++){
        printf("%d ", fibo(i));
    }
}
```

```

    }
}

```

Output:

```

Masukkan jumlah deret= 5
Deret fibonanci dari 5 = 0 1 1 2 3

```

Analisa:

Program tersebut merupakan program untuk menghitung deret balagan fibonanci. Dalam program tersebut terdapat fungsi fibo(int x) digunakan untuk menghitung bilangan fibonaci. Jika x bernilai 0 atau 1 maka akan mengembalikan nilai bilangan x itu sendir, namun jika selain itu maka akan mengembalikan nilai fungsi itu sendiri dengan parameter berubah yaitu mengembalikan nilai fibo(x-2)+fibo(x-1). Fungsi main() digunakan untuk menginputkan jumlah deret fibonaci.Setelah itu fugsi fibo dipanggil menggunakan looping dengan argument l dan menampilkan hasil dari deret bilangan fibonaci.

4. Fungsi rekursi untuk menentukan bilangan prima atau bukan prima

Listing Program:

```

#include <stdio.h>
#include <stdlib.h>

int prime(int number, int index){
    if(index==1){
        return 1;
    }else if(number % index==0){
        return 0;
    }else{
        return prime(number,--index);
    }
}

void main(){
    int num;

    printf("Masukkan bilangan sampai dengan: ");
    scanf("%d", &num);

    printf("Deret bilangan prima: ");
    for(int i=1;i<=num;i++){
        if(prime(i,(int)sqrt(i))){
            printf("%d ", i);
        }
    }
}

```

Output:

```

Masukkan bilangan sampai dengan: 6
Deret bilangan prima: 1 2 3 5

```

Analisa:

Program tersebut merupakan program untuk menampilkan deret bilangan prima dengan batas yang diinputkan. Dalam program ini terdapat fungsi prime(int number,

int index). Jika index bernilai 1 maka akan mengembalikan hasil 1, jika number dibagi habis index maka akan mengembalikan nilai 0, namun jika selain keadaan tersebut akan mengembalikan nilai fungsi itu sendiri dengan parameter (number, --index). Fungsi main() digunakan untuk menginputkan batas dari bilangan prima dengan variabel num. Agar program dapat menampilkan hasil, program menggunakan looping lalu memanggil fungsi prime dengan argument (l,(int)sqrt(i)).

5. Fungsi rekursi untuk menghitung pangkat

Listing Program:

```
#include <stdio.h>
#include <stdlib.h>

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

void main(){
    int x, y;
    printf("Bilangan x pangkat y: ");
    scanf("%d %d", &x, &y);
    printf("%d pangkat %d= %d\n",x,y, pangkat(x,y));
}
```

Output:

```
Bilangan x pangkat y: 3 3
3 pangkat 3= 27
```

Analisa:

Program tersebut merupakan program untuk menghitung bilangan berpangkat. Dalam program tersebut terdapat fungsi pangkat(int x, int y) untuk menghitung bilangan berpangkat. Jika y bernilai 0 akan mengembalikan hasil 1, jika tidak maka akan mengembalikan nilai $x * \text{pangkat}(x, y-1)$. Fungsi main() digunakan untuk menginputkan bilangan yang akan dihitung, bilangan pokoknya diinisialisasi dengan variabel x sedangkan pangkatnya diinisialisasi dengan variabel y. Kemudian fungsi pangkat dipanggil dengan argument (x, y). Lalu menampilkan hasil dengan menggunakan printf.

LATIHAN

1. Buatlah sebuah fungsi yang menulis angka dari n ke 0 dengan menggunakan proses rekursi.

Listing Program:

```
#include <stdio.h>
#include <stdlib.h>

void deretangka(int n){
    if(n<0){
```

```

        return;
    }
    printf("%d ", n);

    deretangka(n-1);
}

void main(){
    int bil;
    int i;
    printf("Masukkan Bilangan: ");
    scanf("%d", &bil);
    printf("Deret Angka %d!: ", bil);
    deretangka(bil);
}

```

Output:

```

Masukkan Bilangan: 5
Deret Angka 5!: 5 4 3 2 1 0

```

Analisa:

Program tersebut adalah program untuk menampilkan deret bilangan dari bilangan ke n sampai 0. Dalam program ini terdapat fungsi `deretangka(int n)` untuk membuat deret bilangan dengan rekursif. Jika n bernilai kurang dari 0 maka akan mengembalikan hasil 0. Jika tidak maka akan mencetak nilai dari n lalu akan memanggil fungsi itu sendiri dengan parameter (n-1). Dalam fungsi `main()` digunakan untuk memasukkan batas dari deret bilangan yang diinisialisasikan dengan `bil`, lalu mencetak bilangan dengan memanggil fungsi `deretangka` dengan argument bilangan.

2. Tuliskan sebuah fungsi untuk menulis angka dari 0 ke n dengan menggunakan proses rekursi.

Listing Program:

```

#include <stdio.h>
#include <stdlib.h>

void deretbalik(int n){

    if(n<0){
        return;
    }

    deretbalik(n-1);

    printf("%d ", n);

}

void main(){
    int bil;
    int i;
    printf("Masukkan Bilangan: ");
    scanf("%d", &bil);
    printf("Deret Bilangan : ", bil);
    deretbalik(bil);
}

```

```
}
```

Output:

```
Masukkan Bilangan: 7
Deret Bilangan : 0 1 2 3 4 5 6 7
```

Analisa:

Program tersebut merupakan program untuk menampilkan deret angka dari batas yang diinputkan dari 0 hingga n (bilangan yang diinputkan). Dalam program ini terdapat fungsi `deretbalik(int n)` digunakan untuk membuat deret bilangan dengan rekursif. Jika n bernilai kurang dari 0 maka akan mengembalikan hasil 0, jika tidak maka memanggil fungsi itu sendiri dengan parameter (n-1) lalu mencetak nilai dari n. Fungsi `main()` digunakan untuk menginputkan batas dari deret bilangan yang diinisialisasikan dengan variabel `bil` dan mencetak nilai dengan memanggil fungsi `deretbalik` dengan argument `bil`.

3. Tuliskan sebuah fungsi rekursi yang melakukan pengecekan apakah sebuah elemen X merupakan anggota dari sebuah array `a[n]`.

Listing Program:

```
#include <stdio.h>
#include <stdlib.h>

int cekbil(int X,int a[], int n){
    if(n==0){
        return 0;
    }
    if(a[n-1]==X){
        return 1;
    }else{
        cekbil(X,a,n-1);
    }
}

void main(){
    int n, a[MAX], X;
    printf("Memasukkan berapa bilangan: ");
    scanf("%d", &n);
    for(int i=0;i<n;i++){
        printf("Masukkan Bilangan ke-%d: ", i+1);
        scanf("%d", &a[i]);
    }
    printf("Cek bilangan= ");
    scanf("%d", &X);
    if(cekbil(X,a,n)==1){
        printf("%d Merupakan anggota dari array", X);
    }else{
        printf("Bukan anggota array");
    }
}
```

Output:

```

Memasukkan berapa bilangan: 5
Masukkan Bilangan ke-1: 45
Masukkan Bilangan ke-2: 23
Masukkan Bilangan ke-3: 75
Masukkan Bilangan ke-4: 12
Masukkan Bilangan ke-5: 66
Cek bilangan= 75
75 Merupakan anggota dari array

```

```

Cek bilangan= 100
Bukan anggota array

```

Analisa:

Program ini adalah program untuk mengecek bilangan dalam array menggunakan rekursif. Dalam program ini terdapat fungsi cekbil(int X, int a[], int n) digunakan untuk mengecek suatu bilang dalam array dengan ketentuan X merupakan bilangan yang akan di cek, a[] merupakan array dari suatu bilangan berderet, dan n merupakan indeks dalam array. Jika nilai dari n bernilai 0 maka akan mengembalikan nilai 0, jika a[n-1] bernilai sama dengan X maka akan mengembalikan hasil 1, jika tidak maka akan memanggil fungsi itu sendiri dengan parameter (X, a, n-1).

Fungsi main() digunakan untuk menginputkan batas dari bilangan yang diinisialisasikan dengan n. Lalu bilangan diinputkan ke dalam array a[] menggunakan looping. Kemudian, menginputkan bilangan yang akan di cek yang diinisialisasikan dengan variabel X, lalu masuk ke dalam pengondisian. Jika fungsi cek(X, a, n) bernilai sama dengan 1 maka bilangan yang diinputkan ada di dalam array a[], jika tidak merupakan bukan dari anggota array a[].

4. Tulis sebuah fungsi yang melakukan pengecekan apakah sebuah angka merupakan bilangan prima atau bukan (n bukan bilangan prima jika dapat dibagi dengan angka kurang dari n)

Listing Program:

```

#include <stdio.h>
#include <stdlib.h>

int prima(int n, int indeks){
    if(indeks==1){
        return 1;
    }else if(n%indeks==0){
        return 0;
    }else{
        return prima(n,--indeks);
    }
}

void main(){
    int bil;
    printf("Masukkan Batas Bilangan\t: ");
    scanf("%d", &bil);

    if(prima(bil,(int)sqrt(bil)))
        printf("=====\n");
}

```



```

printf("%d Merupakan Bilangan Prima",bil);

printf("\n=====\\n");
printf("\\nDeret Bilangan Prima dengan Batas %d\\t: ", bil);
for(int i=1; i<=bil; i++){
    if(prima(i,(int)sqrt(i))){
        printf("%d ", i);
    }
}
printf("\\n");
}else{
printf("=====\\n");
printf("Bukan Merupakan Bilangan Prima");
printf("\\n");
}
}
}

```

Output:

```

Masukkan Batas Bilangan : 7
=====
7 Merupakan Bilangan Prima
=====

Deret Bilangan Prima dengan Batas 7      : 1  2  3  5  7

```

```

Masukkan Batas Bilangan : 8
=====
Bukan Merupakan Bilangan Prima

```

Analisa:

Program tersebut merupakan program untuk menentukan apakah bilangan yang diinputkan termasuk bilangan prima dan menampilkan deret bilangan prima dengan batas yang diinputkan. Dalam program tersebut terdapat fungsi `prima(int n, int index)`. Jika indeks bernilai sama dengan 1 maka akan mengembalikan hasil 1, jika `n` dibagi habis dengan indeks maka akan mengembalikan hasil sama dengan 0, jika tidak termasuk kondisi tersebut maka fungsi memanggil dirinya sendiri dengan parameter `(n, --indeks)`. Fungsi `main()` digunakan untuk menginputkan batas bilangan yang diinisialisasikan dengan variabel `bil`. Lalu mengecek bilangan prima dengan memanggil fungsi `prima(bil, (int)sqrt(bil))`. Lalu jika bilangan prima, deret bilangan prima dengan batas inputan akan dicetak menggunakan looping lalu memanggil fungsi `prima(i, (int)sqrt(i))`.

KESIMPULAN

Rekursif merupakan kemampuan suatu rutin untuk memanggil dirinya sendiri. Kelebihan dari rekursif yaitu lebih mudah dipahami jika sudah memahami cara pemakaiannya, lebih mudah diimplementasikan, dan merupakan solusi yang lebih elegan jika menggunakan rekursif. Sedangkan kekurangan dari rekursif sendiri yaitu memakan banyak memori dikarenakan memanggil fungsi berkali kali, lebih lambat, dan sulit untuk debugging.