**LAPORAN PRAKTIKUM 6**

NAMA : ADITYA PUTRA PRATAMA

NIM : 21091397043

PRODI : D4 MANAJEMEN INFORMATIKA

**D. PENDAHULUAN**

1. Apa yang dimaksud dengan rekursif?

Rekursif merupakan suatu metode perulangan dalam pemrograman yang digunakan untuk memanggil dirinya sendiri secara berulang-ulang. Proses Rekursif memerlukan kondisi untuk membatasi perualangan agar memori yang digunakan untuk menyimpan proses perulangan tidak habis.

1. Tuliskan fungsi untuk menghitung nilai faktorial

Faktorial dari bilangan bulat positif n didefinisikan sebagai berikut :

n! = n x (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)

1. Tuliskan fungsi untuk menampilkan nilai fibonanci dari deret fibonacci

Contoh bilangan fibonacci adalah sebagai berikut :

1 1 2 3 5 8 13 21 34 55 89 …

Bilangan ke-N (N > 2) dalam barisan dapat dicari dari dua bilangan sebelumnya yang terdekat dengan bilangan sebelumnya yang terdekat dengan bilangan N, yaitu bilangan ke-(N-1) dan bilangan ke-(N-2), sehingga dapat dirumuskan sebagai berikut :

Fibonacci(1) = 1 (1)

Fibonacci(2) = 1 (2)

Fibonacci(N) = Fibonacci (N-1) + Fibonacci (N-2) (3)

**D. PERCOBAAN**

**PERCOBAAN 1 :** Fungsi rekursif untuk menghitung nilai factorial

*/\* 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 REKRUSIF"<<endl;*

*cout <<endl;*

*cout <<"Masukkan 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));*

*}*

**PERCOBAAN 2 :** Fungsi rekursi untuk menampilkan deret fibonanci

*/\* Fungsi Rekursif Untuk Menampilkan Deret Fibonachi \*/*

*#include <iostream>*

*using namespace std;*

*int fibonachi (int m){*

*if (m==0||m==-1)*

*{*

*return m;*

*}*

*else*

*{*

*return (fibonachi(m-1) + fibonachi(m-2));*

*}*

*}*

*int main (){*

*int n, m=0;*

*cout <<"Masukkan Batas Deret Bilangan Fibonachi : "; cin >> n;*

*cout <<"Deret Fibonachi : ";*

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

*cout << fibonachi(m)<<" ";*

*m++;*

*}*

*return 0;*

*}*

**PERCOBAAN 3 :** Fungsi rekursi untuk menentukan bilangan prima atau bukan prima

*/\* Fungsi rekursif Untuk Menentukan Bilangan Prima Dan Bukan Bilangan 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 <<"Masukkan Bilangan : "; cin >>bil;*

*if (cek(bil)){*

*cout <<"Bilangan Prima"<<endl;*

*}else{*

*cout<<"Bukan Bilangan Prima"<<endl;*

*}*

*return 0;*

*}*

**PERCOBAAN 4 :** Fungsi rekursi untuk menghitung pangkat

*/\* Fungsi Rekursif Untuk Menghitung Pangkat \*/*

*#include <iostream>*

*using namespace std;*

*long int pangkatrekrusif (int x, int y);*

*int main (){*

*int x,y;*

*cout <<"FUNGSI REKRUSIF UNTUK MENGHITUNG PANGKAT"<<endl;*

*cout <<endl;*

*cout <<"Masukkan Nilai X : "; cin>> x;*

*cout <<"Masukkan Nilai Y : "; cin>> y;*

*cout <<endl;*

*cout<<x<<"Dipangkatkan "<<y<<"="<<pangkatrekrusif(x,y)<<endl;*

*}*

*long int pangkatrekrusif(int x, int y){*

*if(y==0)*

*return 1;*

*else*

*return x \* pangkatrekrusif(x,y-1);*

*}*

**E. LATIHAN**

1. **Program rekursif untuk menghitung segitiga Pascal.**

*/\* Program Rekursif segitiga Pascal \*/*

*#include <stdio.h>*

*#include <math.h>*

*unsigned long int fact(int f)*

*{*

*if (f < 2)*

*{*

*return 1;*

*}*

*else*

*{*

*return f \* fact(f - 1);*

*}*

*}*

*int komb(int q, int r)*

*{*

*return (fact(q) / (fact(r) \* fact(q - r)));*

*}*

*int main()*

*{*

*int i, j, q;*

*printf("masukan n: "); scanf("%d", &q);*

*for (i = 0; i < q; i++)*

*{*

*for (j = 0; j <= i; j++)*

*{*

*printf("%d", komb(i, j));*

*}*

*printf("\n");*

*}*

*return 0;*

*}*

1. **Program secara rekursif, masukkan jumlah N karakter dan cetak dalam semua kombinasi.**

*/\* Program Rekursif Kombinasi \*/*

*#include <iostream>*

*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;*

*}*

1. **Program BinarySearch dengan Rekursif (data tentukan sendiri).**

*/\* Program Rekursif BinarySearch \*/*

*#include <iostream>*

*#include <conio.h>*

*#include <iomanip>*

*using namespace std;*

*int data[13] = {2, 5, 9, 10, 14, 32, 16, 41, 67, 88, 90};*

*int cari;*

*void selection\_sort()*

*{*

*int temp, min, i, j;*

*for(i=0; i<13;i++)*

*{*

*min = i;*

*for(j = i+1; j<11; 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 = 11;*

*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<11; 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<11;x++)*

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

*cout<<endl;*

*binarysearch();*

*getch();*

*return 0;*

*}*

1. **Program rekursif untuk memecahkan permasalahan Menara Hanoi.**

*/\* Program Rekursif menara Hanoi \*/*

*#include <iostream>*

*using namespace std;*

*void MenaraHanoi(int N, char asal, char bantu, char tujuan);*

*int main()*

*{*

*int piringan;*

*cout<< "\nPROGRAM MENARA HANOI\n";*

*cout<< "--------------------\n\n";*

*cout<< "Banyaknya piringan: ";*

*cin >> piringan;*

*cout<< endl;*

*MenaraHanoi(piringan,'A','B','C');*

*return 0;*

*}*

*void MenaraHanoi(int N, char asal, char bantu, char tujuan)*

*{*

*if( N == 1)*

*cout<<"Piringan 1 dari "<<asal<< " ke " << tujuan <<endl;*

*else*

*{*

*MenaraHanoi(N-1,asal,tujuan, bantu);*

*cout<<"Piringan " << N <<" dari " << asal << " ke " << tujuan<<endl;*

*MenaraHanoi(N-1, bantu, asal, tujuan);*

*}*

*}*

1. **Jelaskan proses rekursif untuk program dibawah ini !**

*void decToBin(int num) {*

Deklarasi fungsi void menggunakan parameter fungsi tipe int variabel num.

*if (num > 0)*

*{*

*decToBin(num / 2);*

*cout << num % 2;*

*}}*

Kondisi jika akan berjalan ketika nilai variabel num > 0, maka akan dioperasikan ke dalam decTobin. Tanda { = awal dari fungsi if. *Cout << num%2;}}* menampilkan hasil output nilai variabel num modulu 2 pada hasil program. Tanda } = akhir fungsi void. Tanda } = akhir dari fungsi if. Tanda } = akhir fungsi void decTobin.

1. **Jelaskan proses rekursif untuk program dibawah ini !**

*boolean search(int[] x, int size, int n) {* merupakan deklarasi variabel search dengan tipe data boolean variabel x, size dan n dengan tipe data integer.

*if (size > 0) {* merupakan suatu kondisi jika nilai variabel sixe yang dimasukkan lebih dari nol.

*if (x[size-1] == n) {*

*return true;*

*} else {*

*return search(x, size-1, n);*

*}*

*} return*

*false;*

*}*

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

1. **Jelaskan proses rekursif untuk program dibawah ini !**

*boolean binarySearch(int[] x, int start, int end, int n) {* merupakan deklarasi variabel binarySearch dengan tipe data boolean, variabel x, start, end, dan n dengan tipe data integer.

*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 deklarasi 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) = true. Tanda { = awal dari fungsi if. *} else { if (x[mid] < n) { return search(x, mid+1, end, n);* kondisi (else if) akan berjalan ketika x[mid] < n maka nilai balik (return value) akan dioperasikan ke dalam return search.

1. **Jelaskan proses rekursif untuk program dibawah ini !**

*int mystery(int a, int b) {*  Merupakan deklarasi variabel mystery dengan tipe data integer variabel a dan b.

*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.

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

*int mystery(int a, int b) {* Merupakan pendeklarasian variable mystery dengan tipe data integer variabel a dan 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);*

*}*

*}*

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 (else) maka nilai akan dioperasikan ke deklarasi variabel m1 dan m2 tipe data int. *mystery(a, m1);mystery(m2, b);}}* pendeklarasian variabel mystery di dalam fungsi int mystery. Tanda } adalah akhir dari fungsi int mystery.

1. **Jelaskan proses rekursif untuk program dibawah ini !**

*int f(int n) {* merupakan deklarasi variabel f dengan tipe data integer variabel n.

*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 dan dioperasikan ke dalam return *2\*f(n-2) + f(n-3)*.

1. **Jelaskan proses rekursif untuk program dibawah ini !**

*int square(int n) {* 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) { m*erupakan deklarasi fungsi cube menggunakan parameter formal fungsi tipe data int variabel n.

*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.