Shandy Ilham Alamsyah\_21091397015

**Tugas Analisa Praktikum 6**

**C. Pendahuluan**

1. Apa yang dimaksud dengan rekursif?

Jawaban :

Rekursif adalah suatu proses atau prosedur dari fungsi yang memanggil dirinya sendiri secara berulang-ulang. Karena proses dalam rekursif ini terjadi secara berulang-ulang maka harus ada kondisi yang membatasi pengulangan persebut, jika tidak maka proses tidak akan pernah berhenti sampai memori yang digunakan untuk menampung proses tersebut tidak dapat menampung lagi/penuh.

1. Tuliskan fungsi untuk menghitung nilai faktorial

Jawaban :

long int faktorial (int A){

if (a==1){

return(a);

}else{

return (a\*faktorial(a-1));

}

1. Tuliskan fungsi untuk menampilkan nilai fibonanci dari deret fibonanci

Jawaban :

int fibonacci(int n) {

if (n == 0 || n ==1){

return n;

} else {

return (fibonacci(n-1) + fibonacci(n-2));

}

}

**D. Analisa Percobaan**

* **Percobaan 1** : Fungsi rekursif untuk menghitung nilai faktorial

#include <iostream>

**using namespace** std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file (<iostream>). <iostream> sendiri digunakan untuk mendeklarasikan statement input output.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

**long int** faktorial (**int** A);

* **long int** merupakan variasi dari tipe data integer (**int**) yang berfungsi untuk variabel bilangan bulat yang jumlahnya sangat banyak, dan memiliki fungsi bernama faktorial yang memiliki argument (**int** A) bertipe ‘**int**’ dan bervariabel A.

**int** main(){

* Merupakan pendeklarasian fungsi main yang bertipe data ‘**int**’, dan selalu diikuti tanda kurung ”( )” untuk menandakan sebuah fungsi.

**int** r, hasil;

* Merupakan pendeklarasian variabel ‘r’ dan ‘hasil’ yang bertipe data ‘**int**’.

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

* cout<<"..."; merupakan proses untuk menampilkan output "MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF", dan <<endl berfungsi untuk mengakhiri line/baris (seperti fungsi enter pada keyboard).

cout<<endl;

* Berfungsi untuk mengakhiri line/baris.

cout<<"Masukan Nilai = ";

* cout<<"..."; merupakan proses untuk menampilkan output "Masukan Nilai = ".

cin>>r;

* Merupakan proses untuk menginput nilai ke variabel ‘r’ sesuai tipe datanya.

hasil=faktorial(r);

* Berfungsi untuk menyimpan variabel hasil yang berisi fungsi faktorial() dengan nilai parameter nilai ‘r’ tersebut.

cout<<"Faktorial "<<r<<"!= "<<hasil<<endl;

* cout<<"..." merupakan proses untuk menampilkan output "Faktorial". <<r<< berfungsi untuk menampilkan nilai dari input ‘r’ sebelumnya.
* "!= " merupakan proses untuk menampilkan output "!= ". <<hasil<< berfungsi untuk menampilkan nilai dari input ‘hasil’ sebelumnya.

**long int** faktorial (int A){

**if** (A==1)

return(A);

**else**

return (A\*faktorial(A-1));

}

* Ini merupakan fungsi rekursif faktorial dimana jika nilai ‘A’ yang diinputkan 1 maka nilai balik (*return value*) adalah nilai ‘A’ itu sendiri. Sementara jika tidak maka akan dihitung menggunakan rumus faktorial yaitu (A\*faktorial(A-1)).
* **Percobaan 2** : Fungsi rekursi untuk menampilkan deret fibonanci

#include <iostream>

**using namespace** std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file (<iostream>). <iostream> sendiri digunakan untuk mendeklarasikan statement input output.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

**int** fibonacci(**int** m) {

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

return m;

}**else** {

return (fibonacci(m-1) + fibonacci(m-2));

}

}

* Ini merupakan fungsi rekursif fibonacci dimana jika nilai ‘m’ yang diinputkan 0 atau 1 maka nilai balik adalah nilai ‘m’ itu sendiri. Selain itu akan dihitung menggunakan rumus fibonacci yaitu (fibonacci(m-1) + fibonacci(m-2)).

int main(){

* Merupakan pendeklarasian fungsi main yang bertipe data ‘**int**’, dan selalu diikuti tanda kurung ”( )” untuk menandakan sebuah fungsi.

int n, m= 0;

* Merupakan pendeklarasian variabel ‘n’ dan ‘m’ yang bertipe data ‘**int**’, serta mendeklarasikan bahwa variabel ‘m’ bernilai 0.

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

cin >> n;

cout << "Deret Fibonacci: ";

* cout<<"..."; merupakan proses untuk menampilkan output "Masukan Batas Deret Bilangan Fibonacci : ".
* Merupakan proses untuk menginputkan nilai ke variabel ‘n’ sesuai tipe datanya.
* cout<<"..."; untuk menampilkan output "Deret Fibonacci: ".

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

cout << fibonacci(m) <<" ";

m++;

}

return 0;

}

* Merupakan proses perulangan dimana variabel ‘i’ untuk menyimpan nilai hitungan yang akan dimulai dari 1, dan akan terus bertambah (karena i++) sampai batas dari nilai ‘n’ yang telah diinputkan. Dan akan ditampilkan melalui fungsi fibonacci tiap perulangannya, dan nanti akan kembali ke-0.
* **Percobaan 3** : Fungsi rekursi untuk menentukan bilangan prima atau bukan prima

#include <iostream>

**using namespace** std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file (<iostream>). <iostream> sendiri digunakan untuk mendeklarasikan statement input output.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

**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);

}

}

* Ini merupakan fungsi bernama ambil bertipe data ‘**int**’ yang memiliki argument (**int** bil, i) bertipe ‘**int**’ dan bervariabel bil dan i.
* Pada fungsi ambil ini ada kondisi percabangan (If Else) dimana selama nilai i belum sama dengan 1 maka nilai bil akan terus dilakukan modulus (%) pada nilai i dengan catatan nilai i akan ditambahkan 1 jika hasil bil%i==0 jika kondisi tidak terpenuhi ditambah 0.

**int** cek(int bil){

**if** (bil > 1) {

return (ambil(bil, bil) == 2);

} **else**

return false;

}

}

* Ini merupakan fungsi bernama cek bertipe data ‘**int**’ yang memiliki argument (**int** bil) bertipe ‘**int**’ dan bervariabel bil.
* Pada fungsi cek ini ada kondisi percabangan dimana jika bil lebih dari 1 (karena bilangan prima dimulai dari 2), maka akan diproses pada fungsi ambil().

**int** main(){

* Merupakan pendeklarasian fungsi main yang bertipe data ‘**int**’.

**int** bil;

* Merupakan pendeklarasian variabel bil yang bertipe data integer.

cout<<"Masukan Bilangan : "; **cin**>>bil;

* cout<<"..."; merupakan proses untuk menampilkan output. cin>>bil; merupakan proses untuk menginput nilai variabel bil.

**if** (cek(bil)){

cout<<"Bilangan Prima"<<endl;

}**else** {

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

}

return 0;

}

* Ini merupakan kondisi percabangan dimana jika hasil dari cek tadi bernilai true maka akan ditampilkan output "Bilangan Prima", sedangkan jika hasilnya false maka akan ditampilkan output "Bukan Bilangan Prima".
* **Percobaan 4** : Fungsi rekursi untuk menghitung pangkat

#include <iostream>

**using namespace** std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file (<iostream>). <iostream> sendiri digunakan untuk mendeklarasikan statement input output.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

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

* **long int** merupakan variasi dari tipe data integer yang berfungsi untuk variabel bilangan bulat yang jumlahnya sangat banyak, dan memiliki fungsi bernama pangkatrekursif yang memiliki argument (**int** x,y) bertipe ‘**int**’ dan bervariabel x dan y.

**int** main(){

* Merupakan pendeklarasian fungsi main yang bertipe data ‘**int**’.

**int** x,y;

* Merupakan pendeklarasian variabel ‘x’ dan ‘y’ yang bertipe data ‘**int**’.

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

* cout<<"..."; merupakan proses untuk menampilkan output "FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT", dan <<endl berfungsi untuk mengakhiri line/baris (seperti fungsi enter pada keyboard).

cout<<"Masukan Nilai X = "; cin>>x;

cout<<"Masukan Nilai Y = "; cin>>y;

* cout<<"..."; merupakan proses untuk menampilkan outputnya dan cin>>...; merupakan proses untuk menginput nilai variabel x dan y.

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

* cout<<...<< merupakan proses untuk menampilkan nilai dari variabel tersebut. "..." untuk menampilkan output. <<pangkatrekursif(x,y)akan menampilkan hasil x dipangkatkan y.

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

**if** (y==0)

return 1 ;

**else**

return x \* pangkatrekursif(x,y-1); }

* Ini adalah fungsi dari rekursi perpangkatan dengan kondisi, jika nilai y yang dimasukan adalah 0 maka akan di set nilai baliknya adalah 1, namun jika tidak maka fungsi tersebut di set nilai baliknya dimana nilai x akan dikalikan nilai y-1.

**E. Latihan**

1. Jelaskan program rekursif untuk menghitung segitiga Pascal !

#include <iostream>

#include <conio. h>

**using namespace** std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file. <iostream> sendiri digunakan untuk mendeklarasikan statement input output. Dan <conio> digunakan untuk fungsi getch();.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

**long** faktorial(**int** n) {

**long** z = 1;

**int** i = 1;

* **long int** merupakan variasi tipe data yang berfungsi untuk variabel yang nilainya berjumlah sangat banyak, dan memiliki fungsi bernama faktorial yang memiliki argument bertipe ‘**int**’ dan bervariabel n.
* Merupakan pendeklarasian variabel ‘z’ yang bertipe data ‘**long**’ dan ‘i’ yang bertipe data ‘**int**’ dan bernilai 1.

**while**(i<=n){

z=z\*i;

i++;}

**return** z;

* ini merupakan proses perulangan dengan nilai i akan terus bertambah sampai dengan nilai maks sesuai nilai ‘n’ yang diinputkan, lalu akan memiliki nilai balik z, yang akan diproses dengan z=z\*i.

int main() {

int i, j;

* Merupakan pendeklarasian fungsi main yang bertipe data ‘**int**’. Juga pendeklarasian variabel ‘i’ dan ‘j’ yang bertipe data ‘**int**’.

cout<<" ========================= "<<endl;

cout<<"| SEGITIGA PASCAL |"<<endl;

cout<<" ========================= "<<endl<<endl;

* cout<<"..."; merupakan proses untuk menampilkan output sesuai "..." .

for (i=0; i<6; i++) {

for (j=0; j<6-i-1; j++){

cout<<" ";

}

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

cout<<faktorial(i)/(faktorial(j)\*faktorial(i-j))<<" ";

}

cout << endl;

}

* Ini merupakan program pengulangan bersarang nested loop, dimana nilai akan terus bertambah sampai <6, kemudian akan dioutputkan sesuai rumus faktorial(i)/ (faktorial(j)\*faktorial(i-j)), dan diakhiri dengan <<endl untuk mengakhiri line.

1. Jelaskan program secara rekursif, masukkan jumlah N karakter dan cetak dalam semua kombinasi !

#include <iostream>

#include <conio.h>

using namespace std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file. <iostream> sendiri digunakan untuk mendeklarasikan statement input output. Dan <conio> digunakan untuk fungsi getch();.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

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);

}

}

* Ini merupakan program fungsi dari rekursi kombinasi, yang didalamnya terdapat nilai yang akan digunakan untuk membuat rangkaian rekursif kombinasi.

void printAllKLength(char set[], int k,int n)

{

printAllKLengthRec(set, "", n, k);

}

* Ini merupakan pendeklarasian fungsi dan beberapa variabel, dan juga program untuk mengatur panjang rangkaian kombinasi tersebut.

int main(){

* Merupakan pendeklarasian fungsi main yang bertipe data ‘**int**’.

**char** ch[] = {'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;

* Ini Pendeklarasian variabel ‘ch’ yang bertipe data ‘char’ dan berisi abjad a-z serta variabel ‘N’ yang bertipe data ‘**int**’;

cout<<"\njumlah karakter : "<<' ';

* cout<<"..."; merupakan proses untuk menampilkan output

while(cin >> N){

if(N != -1){

printAllKLength(ch,N, N) ;

}

else if(N>26){

cout<<"karakter yang anda masukkan salah!";

break;

}

else{

break;

}

}

* Ini merupakan keterangan dari sebuah program kombinasi. Didalamnya terdapat jumlah karakter yang akan digunakan dalam suatu programm kombinasi. Dan apabila jumlahnya lebih dari 26 maka akan ada output tersebut.

1. Jelaskan program BinarySearch dengan Rekursif !

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#define size 10

using namespace std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file. <iostream> sendiri digunakan untuk mendeklarasikan statement input output. <stdio.h> digunakan untuk operasi standar input output. <stdlib.h> digunakan untuk operasi pembanding dan operasi konversi.
* Script #define size 10 artinya membuat konstanta bernama size yang diisi nilai 10.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

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);

}

}

* Ini merupakan algoritma fungsi binary search dan menggunakan if else, dimana akan dibagi 2 elemen yaitu low dan high. Jika nilai x=a[mid]akan dikembalikan ke nilai mid. Jika x<a[mid] maka akan dimasukkan ke (a, x, low, mid-1);. Selain itu akan dimasukkan ke(a, x, mid + 1, high).

**int** main() {

**int** num, i, key, position;

**int** low, high, list[size];

* Ini merupakan pendeklarasian fungsi ‘main’ bertipe data ‘**int**’ dan pendeklarasian beberapa variabel bertipe data ‘**int**’

cout<<"\nMasukan Jumlah Besar data:"; cin>>num;

cout<<"\nMasukan Data:\n";

for (i=0; i<num; i++) {

cin>>list[i];

}

* Merupakan beberapa output dan input untuk memasukkan data data yang nantinya bisa dicari.

low = 0;

high = num - 1;

cout<<"\nMasukan data yang ingin di cari : "; cin>>key;

position = binsearch(list, key, low, high);

if (position != -1) {

cout<<"\nData "<<key<<" berada pada Index ke - "<<position<<endl;

} else

cout<<"\nData tidak di temukan"<<endl;

return (0);

}

* Pada script ini akan diinput data yang akan dicari, kemudian akan diproses posisinya dengan command position = binsearch(list, key, low, high);, kemudian akan ditampilkan datanya jika ditemukan dan akan ditampilkan cout<<"\nData tidak di temukan"<<endl; jika tidak ada datanya.

1. Jelaskan program rekursif untuk memecahkan permasalahan Menara Hanoi !

#include <stdio.h>

#include <iostream>

using namespace std;

* Script #include merupakan *prepocessor directive* yang merupakan petunjuk bagi compiler untuk menyisipkan file lain berupa header file. <iostream> sendiri digunakan untuk mendeklarasikan statement input output. <stdio.h> digunakan untuk operasi standar input output.
* Using namespace std; digunakan untuk menghindari kesalahan berupa tidak dikenalinya cout dan cin oleh sistem.

void Hanoi(int n, char dari\_pasak, char ke\_pasak, char pasak\_tmp){

if (n == 1){

cout<<"\nPindahkan disc 1 dari pasak "<<dari\_pasak<<" ke pasak "<<ke\_pasak;

return;

}

Hanoi(n-1, dari\_pasak, pasak\_tmp, ke\_pasak);

cout<<"\nPindahkan disc "<<n<<" dari pasak "<<dari\_pasak<<" ke pasak "<<ke\_pasak;

Hanoi(n-1, pasak\_tmp, ke\_pasak, dari\_pasak);

}

* Script ini merupakan pengaturan dari sebuah program menara hanoi agar disc dapat berpindah dari pasak ke pasak sesuai dengan yang diperintahkan.

int main(){

int n;

cout<<"Masukan Jumlah Disc : ";cin>>n;

Hanoi(n, 'A', 'C', 'B');

return 0;

}

* Script ini merupakan program untuk memasukkan jumlah disc yang akan dipindahkan dari pasak ke pasak tersebut.

1. Jelaskan proses rekursif untuk program dibawah ini !

void decToBin(int num)

{

if (num > 0)

{

decToBin(num / 2);

cout << num % 2;

}

}

* Merupakan fungsi untuk mengubah decimal ke biner, dengan menggunakan kondisi if else, dimana num akan dibagi 2 dan akan ditampilkan modulus 2, kemudian sisanya akan disimpan di array.

1. 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;

}

* Script ini merupakan fungsi rekursif untuk pencarian nilai, dengan kondisi percabangan dimana nilai balik akan dioperasikan ke dalam return search bernilai false.

1. 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);

}

}

}

* Script ini merupakan fungsi binary search, dengan kondisi percabangan jika nilai x[mid] bernilai 0 maka nilai baliknya benar, sedangkan apabila x[mid] kurang dari n maka akan dicari dengan rumus return search(x, start, mid-1, n);.

1. Jelaskan proses rekursif untuk program 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;

}

* Script ini merupakan fungsi dengan kondisi percabangan apabila nilai b yang dimasukkan bernilai 1 maka nilai baliknya adalah 0.

1. 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);

}

}

* Script ini merupakan fungsi dengan kondisi percabangan, jika nilai a = nilai b akan ditampilkan nilai variabel a, sedangkan kondisi else maka nilai akan dioperasikan kedeklarasi m1 dan m2.

1. 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);

* Script ini merupakan fungsi dengan kondisi percabangan jika nilai n=0 maka nilai baliknya adalah 0, dan apabila n=1 maka nilai baliknya adalah 1, tapi jika nilai n=2 maka nilai baliknya adalah 1 dan akan dimasukkan ke operasi return 2\*f(n-2) + f(n-3);.

1. 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;

}

* Script ini merupakan fungsi square dengan kondisi percabangan jika n=0, maka nilai baliknya 0 dan akan dimasukkan ke rumus square(n-1) + 2\*n - 1;.
* Script ini merupakan fungsi cube dengan kondisi percabangan jika n=0, maka nilai baliknya 0 dan akan dimasukkan ke rumus

return cube(n-1) + 3\*(square(n)) - 3\*n + 1.