LAPORAN HASIL ANALISA

**“ DARI TUGAS PRAKTIKUM KE-6 MATA KULIAH PEMROGRAMAN DASAR C,D, DAN E “**



**DI SUSUN OLEH :**

**FAHREZA DZAKY RAHMATULLAH**

**21091397061**

**PROGRAM STUDI D IV MANAJEMEN INFORMATIKA**

**PROGRAM VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2021/2022**

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Allah SWT, Tuhan Yang Maha Esa yang telahmemberikan rahmat serta hidayah-Nya sehingga penyusunan tugas ini dapat diselesaikan.Tugas ini disusun untuk diajukan sebagai tugas mata kuliah Pemrograman Dasar dengan judul Laporan Analisa Diploma 4 Universitas Negeri Surabaya Jurusan D4 Manajemen Informatika.

Terima kasih disampaikan kepada Bapak Andy selaku dosen mata kuliah Pemrograman Dasar yang telah membimbing dan memberikan kuliah demi lancarnya tugas ini.Demikianlah tugas ini disusun tentunya Kami masih banyak kekurangan dan kelemahan dalam mengerjakan tugas ini, untuk itu kami mengharap kritik dan saran untuk memperbaiki kesalahan dalam pengerjaan tugas selanjutnya, Terima kasih.

**HASIL DAN PEMBAHASAAN**

Dari modul praktikum 6 untuk mata kuliah pemrograman dasar yang saya pelajari saya Analisa dan saya kerjakan berbagai tugasnya, saya dapat menganalisanya yang akan ketik di laporan ini yang akan saya urutkan dari mulai tugas yang berada bagian C sampai tugas yang berada di bagian E, maka dari itu izinkan saya membuat laporan ini sebaik mungkin untuk memenuhi tugas dari pak andy yang telah diberikan kepada saya.

Di bagian bawah ini saya menjelaskan hasil dari dari Analisa tugas bagian C yang dapat saya Analisa yaitu sebagai berikut:

**1. HASIL ANALISA SAYA PADA BAGIAN TUGAS C**

**C.** Tugas Pendahuluan

1. Rekursif yaitu suatu proses atau prosedur dari fungsi yang memanggil dirinya sendiri secara berulang-ulang

2. Di bawah ini adalah hasil kode yang yang saya anailsa

#include <iostream>

**using** **namespace** std;

**int** hitungFaktorial(**int** input) {

**if**(input > 1)

**return** input \* hitungFaktorial(input - 1);

**else**

**return** 1;

}

**int** main()

{

  cout << "||  Fungsi Hitung Faktorial ||" << endl;

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

  cout << endl;

**int** angka;

  cout << "Input angka: ";

  cin >> angka;

  cout << angka << "! = " << hitungFaktorial(angka);

  cout << endl;

**return** 0;

3. Yang ketiga adalah seperti yang ada dibawah ini:

#include <iostream>

using namespace std;

int fibonacci(int n) {

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

return n;

} else {

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

}

}

int main() {

int n, i, j = 0;

cout << "Masukkan batas jumlah bilangan fibonacci: ";

cin >> n;

cout << "Hasil bilangan fibonacci: \n";

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

cout << "fibonacci ke-"<< i <<": " << fibonacci(j) << endl;

j++;

}

return 0;

}

**HASIL ANALISA BAGIAN D**

Di bagian selanjutnya ini saya menjelaskan hasil Analisa saya dari tugas bagian D yang mencakup kode kode percobaan dari fungsi rekursif untuk digunakan di suatu permasalahan tertentu. Berikut ini adalah hasil Analisa saya.

**2.** **HASIL ANALISA DI PERCOBAAN 1 D**

Analisa Percobaan Pertama kepasa fungsi rekursif untuk menghitung nilai factorial yaitu yang akan saya jelaskan disini yaitu hasil nalisa saya dapat menjelaskan bahwa di dalam fungsi main() pada baris ke 20-21 itu menjelaskan bahwa kode program yang diulis akan meminta inputan user, yaitu angka yang akan dihitung faktorialnya. Angka tersebut akan disimpan ke variable angka.

Kemudian di baris 23 itu terdapat pemanggilan fungsi **hitungfaktorial(angka).** Disini **angka** dikirim sebagai argument dari fungsi **hitungFaktorial()**

**Definisi** fungsi **countFactorial()** ada di baris 5-10. Fungsi ini menerima nilai **input sebagai** parameter input. **Ada kondisi if** dalam fungsi **computeFaktorial()** untuk **melihat** apakah parameter input lebih **besar** dari **1. Dalam hal ini,** jalankan perintah return input \* **countFaktorial (input-1). Ini memiliki arti** rekursif, **dan** fungsi ini memanggil dirinya sendiri. Jika kondisi **if (input>** 1) **salah,** maka fungsi **countFaktorial()** mengembalikan angka **1.**

**3. HASIL ANALISA DI PERCOBAAN 2 D**

Di bagian ini terdapat percobaan kode untuk menampilkan deret Fibonacci disini saya dapat menejelaskan bahwa **Definisi** fungsi **countFactorial()** ada di baris 5-10. Fungsi ini menerima nilai **input sebagai** parameter input. **Ada kondisi if** dalam fungsi **computeFactorial()** untuk **melihat** apakah parameter input lebih **besar** dari **1. Dalam hal ini,** jalankan perintah return input \* **countFactorial (input-1). Ini memiliki arti** rekursif, **dan** fungsi ini memanggil dirinya sendiri. Jika kondisi **if (input>** 1) **salah,** maka fungsi **countFactorial()** mengembalikan angka **1.**Baris 5: **Buat** fungsi Variabl **Fibonacci** yang **merupakan tipe** integer dan menerima **input dari** integer yang **disebut Variable** n. Baris 6-7: **Periksa kondisi. Jika** nilai **Variable** n adalah 0 atau **1,** nilai **n dikembalikan.** Baris 8-9: Jika **nilai Variable** n selain 0 dan **1, lakukan** operasi **rekursif. Operasi** rekursif **menambahkan** hasil **dari** operasi n1 **Fibonacci ke** hasil **dari n2** operasi **Fibonacci.** Baris 15: **Deklarasikan Variable** n, I, **j. Di sini Variable** n digunakan untuk **membatasi** jumlah **angka Fibonacci** yang akan ditampilkan, **Variable** I digunakan untuk **perulangan,** dan **Variable** j digunakan untuk memasukkan nilai **secara berurutan** ke **dalam** fungsi **Fibonacci.** Baris 17-18: **Masukkan** nilai **untuk Variable** n. Baris 20-14: **Mengulangi hingga** nilai **n, memasukkan** nilai j **ke dalam** fungsi **Fibonacci, dan** menampilkan hasil operasi **Fibonacci.**

**4. HASIL ANALISA PERCOBAAN 3 D**

Pada Program ini terdapat variable bil menampung bilangan yang akan diinput oleh user

if (i == 1) {  
return 1;  
}  
else if (bil % i == 0) {  
return 1 + ambil(bil, --i);  
} else {  
return 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.

if (bil > 1) {  
return (ambil(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.

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

**5. HASIL ANALISIS PERCOBAAN 4 D**

Pada contoh program fungsi rekursi untuk menghitung pangkat pada percobaan ke – 4 yaitu dapat saya analisis yaitu user dapat memasukkan nilai x dan y, lalu kemudian x dipangkatkan dengan nilai y.

Jika diliat di fungsi rekrusifnya

long int pangkatrekursif(int x, int y){ if (y==0) return 1 ; else return x \* pangkatrekursif(x,y-1);

Disitu user membaut fungsi dengan nama pangkat rekrusif dan dengan menggunakan 2 parameter yaitu nilai x dan y. jika nilai y yang dimasukkan itu adalah 0 maka akan di set nilai baliknya adalah 1, namun jika tidak itu maka fungsi tersebut di set nilai sebaliknya dimana nilai x akan dikalikan dengan nilai y-1

**LATIHAN E**

**6. HASIL KODE PROGRAM MENGHITUNG SEGITIGA PASCAL**

# include <iostream>

using namespace std;

long int faktorial (int n);

long z = 1;

int i = 1;

while(i<=n){

z=z\*i;i++}

int main()

int a, i, j;

cout<<"Masukan nilai= ";

cin>> a;

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

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

cout<< " ";

}

cout<<endl;

}

return 0;

}

. Int pascal(int row, int col)

if (col == 0 || col == row)

{

return 1;

}

else

{

return pascal(row - 1, col - 1) + pascal(row - 1, col);

}

}

// Script diatas merupakan bagian dari rumus untuk pengerjaan suatu program. Script ini menunjukkan penyusunan suatu output dari suatu program.

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;

}

// Script diatas menjelaskan pengaturan dari sebuah program. Setealah terdapat rumus, maka di script ini lah yang akan diatur penulisan outpunya.

**7. HASIL KODE PROGRAM MEMASUKKAN NILAI N DAN CETAK DALAM SEMUA KOMBINASI**

#include <bits/stdc++.h>

using namespace std;

void printLebarChar(char set[], string huruf,int n, int k)

{

if (k == 0)

{

cout << (huruf) ;

cout << " ";

return;

}

for (int i = 0; i < n; i++)

{

string newHuruf;

newHuruf = huruf + set[i];

printLebarChar(set, newHuruf, n, k - 1);

}

}

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

{

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

}

int main()

{

char set1[] = {'a', 'b', 'c'};

int k = 3;

cout << "Jumlah karakter :" ;

cin >> k;

inputLebarChar(set1, k, 3);

}

**Analisis Program :** Dalam program ini saya membuat 1 fungsi rekursif yaitu kombinasi dengan 3 parameter dimana yang berfungsi untuk mengacak huruf yang diinputkan oleh user, dimana jika jumlah input yang dimasukkan 0 maka tidak muncul huruf yang diacak, jika selain 0 program akan berjalan mengacak jumlah huruf yang diinputkan oleh user, cara kerjanya user input jumlah karakter dan akan dieksekusi oleh program kombinasi dan akan diacak sampai selesai

**8. HASIL KODE PROGRAM BINARY SEARCH**

#include <iostream>

#include <conio.h>

#include <iomanip>

using namespace std;

int data[13] = {1, 6, 2, 3, 10, 17, 14};

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<<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<7;x++)

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

cout<<endl;

binarysearch();

\_getche();

return 0;

}

int data[7] = {1, 8, 2, 5, 4, 9, 7};

int cari;

// Script program diatas berisi banyak data angka biner untuk diurutkan.

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;

}

}

// Script program diatas berisi perumusan angka biner dalam suatu program yang akan dijalankan.

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";

}

// Script program diatas berisi pengaturan penempatan angka biner dalam suatu program untuk dijalankan sesuai dengan inoutnya.

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;

}

// Script program diatas menjelaskan tentang pengaturan program untuk rangkaian angka biner. Dimana data akan dimasukkan, lalu data akan diurutkan, dan yang terakhir data akan ditampilkan sesuai dengan inputnya.

**9. HASIL PROGRAM MENGHITUNG MENARA HANOI**

using namespace std;

void menaraHanoi(int n, char dari\_pasak, char ke\_pasak, char tambah\_pasak)

{

if (n == 1)

{

cout << "Pindahkan disc 1 dari pasak " << dari\_pasak <<" ke pasak " << ke\_pasak<<endl;

return;

}

menaraHanoi(n - 1, dari\_pasak, tambah\_pasak, ke\_pasak);

cout << "Pindahkan disc " << n << " dari pasak " << dari\_pasak <<" ke pasak " << ke\_pasak << endl;

menaraHanoi(n - 1, tambah\_pasak, ke\_pasak, dari\_pasak);

}

int main()

{

int n;

cout <<"Jumlah disk : ";

cin >> n;

cout << "Langkah-langkah nya adalah dengan :" <<endl;

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

return 0;

}

**Analisis Program :**

* Variabel “pasak1” berperan sebagai tiang a, variabel “pasak2” sebagai tiang 'b' sedangkan variabel “media” berperan sebagai tiang 'c'.
* Penjelasan Fungsi pada Hanoi.  
  **Apabila Banya Disk yang di inputkan = 0** maka :  
  Disk akan mengembalikan nilai dari disk itu sendiri
* **Apabila tidak sama dengan 1** maka :  
  Pindahkan n-1 Disk yang lebih kecil dari A ke B  
  Pindahkan Disk yang tersisa dari A ke C.  
  Pindahkan n-1 Disk dari B ke C.
* Pindah pada fungsi main berfungsi untuk memanggil fungsi rekursif dari pindah.

**10. HASIL ANALISA TERHADAP PROGRAM NOMOR 5**

Program ini yaitu program untuk membuat bilangan decimal ke bilangan binner Langkah langkahnya yaitu:

1. Bagi angka dengan 2 hingga % (operator modulus) dan simpan sisanya dalam array

2. Bagi angka dengan 2 melalui / (operator divisi)

3. Ulangi langkah 2 sampai jumlahnya lebih besar dari nol

**11. HASIL ANALISA PROGRAM NOMOR 6**

Program ke 6 yaitu contoh program untuk mencari tipe data dari nilai Boolean

Proses Rekursif suatu program

Boolean search(int [] x, int size, int n) { // menjelaskan tentang ukuran dari sebuah persamaan n dengan boolean.

If (size > 0) { // menjelaskan ukuran dari sebuah bilangan n.

If (x[size -1] == n) {

return true;

} else {

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

}

} return

false;

}

**12. HASIL ANALISA PROGRAM 7**

Dari hasil Analisa yang saya lakukan saya dibawah ini saya jelaskan

Proses rekrusif suatu program

boolean binarySearch {int[] x, int start, int end, int n) { // menghitung angka biner menggunakan boolean.

if (end < start)

return false;

int mid = (start+end) / 2; // hasil awal dan akhir dijumlah dan dibagi dengan 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);

**13. HASIL ANALISA PROGRAM 8**

Dari hasil analisa yang saya dapatkan saya bisa meyimpulkan bahwa

Program rekursif suatu program

Mystery(2, 25) and mystery(3, 11)! // program mencari data nilai

Int mystery(int a, int b) {

If (b == 0);

Return 0;

If (b % 2 == 0)

Return mystery (a+a, b/2); //jumlah nilai akan dijumlah dan dibagi 2.

Return mystery (a+a, b/2) + a; //hasil dari perhittungan di dalam kurung akan di tambah a

**14. HASIL ANALISA PROGRAM 9**

Dari Analisa yang saya lakukan saya dapat menjelaskan bahwa

Program rekursif suatu program

Int mystery(int a, int b) {

If ( a == b) cout<<a<<endl;

Else {

Int m1 = (a + b) / 2; // nilai a dan b akan dijumlah dan dibagi 2

Int m2 = (a + b +1) /2; // nilai a dan b dijumlah dengan 1 lalu dibagi 2

Mystery(a, m1);

Mystery(m2, b);

}

}

**15. HASIL ANALISA PROGRAM 10**

Dari Analisa yang dapat saya jelaskan di bawah ini yaitu

Program rekursif suatu program

Int f (int n) { // program nilai 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

**16. HASIL ANALISA PROGRAM 11**

Hasil analisa yang saya dapatkan di program ini adalah

Cube(5), cube (123)? // membuat program dengan 5 nilai yang berbeda

Int square(int n) {

If (n == 0)

Return 0;

Return square (n-1) + 2\*n – 1; // nilai akan dikurang 1 ditambah 2 dan nilai lainnya akan dikurang 1.

}

Int cube(int n) {

If (in == 0) return 0;

Return cube (n-1) + 3\*square(n)) - 3\*n + 1;

}