

Laporan Resmi
Penerapan Fungsi Rekursif Pada Pemrograman C++



Di susun Oleh :
Rendi Nicolas Mahendra / 21091397071

D4 Manajemen Informatika
Fakultas Vokasi
Universitas Negeri Surabaya
2021

BAB I

PENDAHULUAN

Rekursif merupakan suatu proses atau prosedur dari sebuah fungsi untuk memanggil dirinya sendiri secara repetitif atau berulang. Karena proses pada rekursif dilakukan secara terus-menerus maka harus ada suatu kondisi yang membatasi proses tersebut, karena proses tersebut akan sangat memakan penggunaan resource memori yang digunakan untuk menampung proses tersebut tidak dapat menampung lagi atau penuh.

Keunggulan dari fungsi rekursif ini adalah program yang ditulis menjadi lebih singkat. Karena pada beberapa kasus program rekursif ini sering digunakan seperti, Nilai Faktorial, Segitiga Pascal, Fibonacci, dll. Sementara Kelemahan dari fungsi rekursif ini adalah memakan resource memori yang terlalu besar untuk media penyimpanan sehingga hal tersebut akan menghambat proses. Rekursif sering kali terjadi karena programnya yang repetitif sehingga memori akan terpakai habis dan program bisa hang secara tiba-tiba.

1.1 RUMUSAN MASALAH

- A. apa yang dimaksud dengan Rekursif ?
- B. Tuliskan Fungsi untuk menghitung nilai Faktorial ?
- C. Tuliskan fungsi untuk menampilkan nilai Fibonacci dari suatu deret Fibonacci ?

1.2 Tujuan

- A. Dapat mendefinisikan tentang pengertian dari Rekursif
- B. Mampu menulis program fungsi nilai faktorial dalam bahasa C++
- C. Mampu menulis program fungsi untuk menampilkan bilangan Fibonacci
- D. Mampu menganalisa potongan blok code secara sederhana

BAB II

PEMBAHASAN

2.1 PEMBAHASAN RUMUSAN MASALAH

A. Pengertian Rekursif

Rekursif adalah suatu proses yang memanggil dirinya sendiri. Dalam rekursif sebenarnya terkandung pengertian prosedur atau fungsi. Perbedaannya adalah bahwa rekursif bisa memanggil ke dirinya sendiri, tetapi prosedur atau fungsi harus dipanggil lewat pemanggil prosedur atau fungsi. Pemanggilan diri sendiri dilakukan berulang-ulang sampai mencapai titik Tertentu.

B. Program Fungsi Rekursif Nilai Faktorial

```
#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));

}
```

C. Menulis program bilangan Fibonacci menggunakan Fungsi Rekursif

```
#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;
}

```

2.2 PEMBAHASAN ANALISA PROGRAM

A. Pembahasan Mengenai Program Segitiga Pascal

```

#include <iostream>
using namespace std;

long faktorial(int n) {

    long z = 1;
    int i = 1;

    while(i<=n) {
        z=z*i;
        i++;
    }

    return z;
}

```

```

}

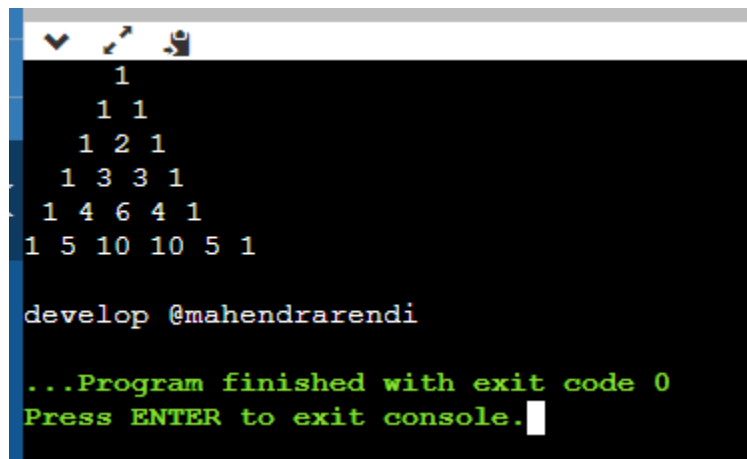
int main() {
    int i, x;

    for (i=0; i<6; i++) {
        for (x=0; x<6-i-1; x++){
            cout << " ";
        }

        for (x=0; x<=i; x++){
            cout << faktorial(i) / (faktorial(x) * faktorial(i - x)) << " ";
        }

        cout << endl;
    }
    cout << "\ndevelop @mahendrarendi";
    return 0;
}

```



```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

develop @mahendrarendi

...Program finished with exit code 0
Press ENTER to exit console.

```

Analisa : Program tersebut berfungsi untuk mencetak segitiga pascal dengan cara rekursif ,Terlihat di situ fungsi untuk segitiga pascal memanggil fungsinya sendiri untuk mereturn hasil dari penjumlahan pascal,dengan rumus $\text{baris}-1, \text{posisi} + \text{SegitigaPascal}$ $\text{baris}-1, \text{posisi}-1$, jadi baris sebelumnya dan posisi sebelumnya di tambah dengan baris sebelumnya dengan posisi di kurangi 1`

B. Pembahasan Mengenai Program Fungsi Kombinasi

```
#include<iostream>
using namespace std;

void printAllKLengthRec(char set[], string prefix, int n, int j)
{
    if (j == 0)
    {
        cout << (prefix) << " ";
        return;
    }
    for (int i = 0; i < n; i++)
    {
        string newPrefix;
        newPrefix = prefix + set[i];
        printAllKLengthRec(set, newPrefix, n, j - 1);
    }
}

void printAllKLength(char set[], int j,int n)
{
    printAllKLengthRec(set," ", n, j);
}

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'};
    char x ;
    int N;

    awal:

    cout << "Jumlah Karakter:" << "" ;
    while(cin >> N){
        if(N != -1){
            printAllKLength(set1,N, N)  ;
        }
        else if(N>26){
            break;
        }
        else{
            break;
        }
        cout << " " << endl;
```

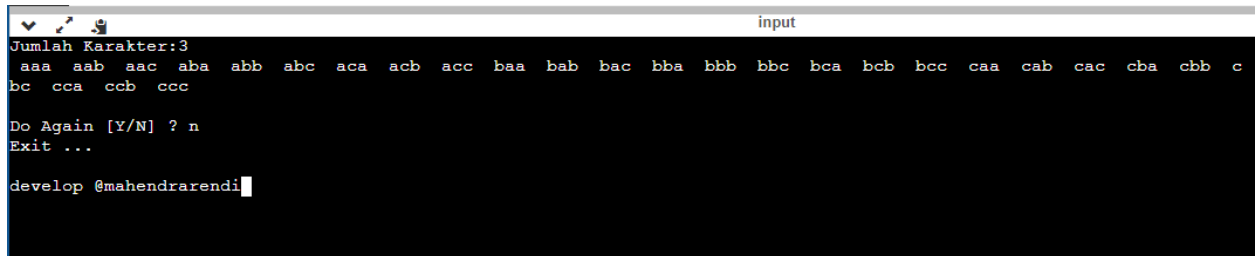
```

        cout << "\nDo Again [Y/N] ? "; cin >> x ;
        if (x == 'y' || x == 'Y'){
            goto awal;
        }
        else if(x == 'n' || x == 'N'){
            cout << "Exit ..." << endl;
        }
        cout << "\ndevelop @mahendrarendi";

    }

    return 0;
}

```



```

input
Jumlah Karakter:3
aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb c
bc cca ccb ccc
Do Again [Y/N] ? n
Exit ...
develop @mahendrarendi

```

Analisa : Program tersebut merupakan suatu program yang di gunakan untuk melakukan kombinasi huruf dengan memanfaatkan perulangan dan menyimpan data kombinasi pada suatu array dan akan di tampilkan sebnayak inputan yang kita berikan.

3. Pembahasan Program Binnary Search

```

#include <iostream>

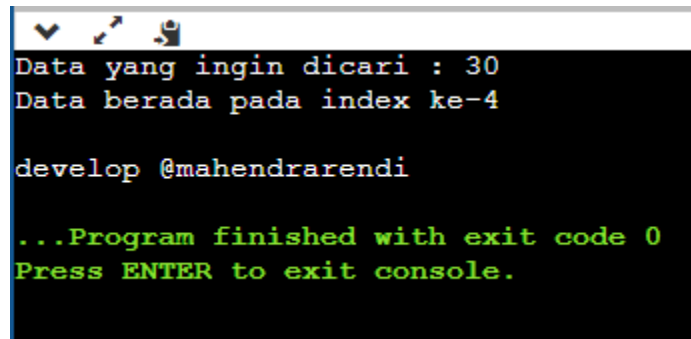
int main() {
    int array_a[10]={10,15,20,25,30,35,40,45,50,55};
    int i, search, flag=0;

    printf("Data yang ingin dicari : ");
    scanf("%i",&search);
    for (i=0;i<10;i++) {
        if (array_a[i]==search) {
            flag=1;
            break;
        }
    }
    if (flag==1) printf(" Data berada pada index ke-%i",i);
}

```

```
else printf("data yang anda cari tidak ditemukan");

printf("\n ");
printf("\ndevelop @mahendrarendi");
}
```

A screenshot of a terminal window with a black background and white and green text. The text shows the program's execution: it prompts for a search value (30), finds it at index 4, displays the developer's name, and then shows a green message indicating the program finished successfully with exit code 0.

```
Data yang ingin dicari : 30
Data berada pada index ke-4

develop @mahendrarendi

...Program finished with exit code 0
Press ENTER to exit console.
```

Analisa : Program tersebut merupakan program yang di gunakan untuk mencari suatu data. Dimana data yang di cari di simpan pada suatu array dan menampilkan urutan data tersebut pada sebuah array yang di simpan pada variabel search

4. Pembahasan Mengenai Program Menara Hanoi

```
#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 i, x;

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



```

    for (x=0; x<6-i-1; x++){
        cout << " ";
    }

    for (x=0; x<=i; x++){
        cout << faktorial(i) / (faktorial(x) * faktorial(i - x)) << " ";
    }

    cout << endl;

}
cout << "\ndevelop @mahendrarendi";
return 0;

}

```

```

Jumlah Disk = 3

Pindahkan Disk 1 dari pasak A ke pasak C
Pindahkan Disk 2 dari pasak A ke pasak B
Pindahkan Disk 1 dari pasak C ke pasak B
Pindahkan Disk 3 dari pasak A ke pasak C
Pindahkan Disk 1 dari pasak B ke pasak A
Pindahkan Disk 2 dari pasak B ke pasak C
Pindahkan Disk 1 dari pasak A ke pasak C

develop @mahendrarendi

...Program finished with exit code 0
Press ENTER to exit console.

```

Analisa : Program tersebut merupakan bentukan dari program mencari nilai faktorial dimana kita memasukan jumlah data yang akan kita hitung menggunakan rumus faktorial $\text{faktorial}(i) / (\text{faktorial}(x) * \text{faktorial}(i - x))$ dengan menggunakan variabel i dan x untuk melakukan perulangan sehingga dapat menghasilkan output yang sesuai dengan apa yang kita perintahkan pada program tersebut.

5. Jelaskan proses rekursif untuk program dibawah ini !

```

void decToBin(int num)
{
    if (num > 0)

```

```
{
decToBin(num / 2); cout << num % 2;
}
}
```

Analisa : Blok code di atas merupakan suatu fungsi dalam melakukan konversi bilangan dari desimal ke biner dengan rumus bilangan / 2 serta modular 2 dengan menggunakan fungsi decToBin

6. 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;
}
```

Analisa : Program tersebut merupakan suatu program fungsi rekursif yang di implementasikan dalam kasus binary search dimana data boolean yang akan di cari yang di simpan pada sebuah elemen array.

7. 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);
        }
    }
}
```

Analisa : Program di atas merupakan suatu program fungsi rekursif yang di implementasikan pada suatu kasus binary search dengan mencari data median yang di

simpan pada suatu array dengan menggunakan rumus $((start+end)/2)$ pada kode program di atas.

8. Jelaskan proses rekursif untuk program dibawah ini !

```
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;  
}
```

Analisa : Merupakan suatu program dengan rekursif fungsi mystery yang di simpan pada variabel a dan b dengan type data integer. Jika variabel B bernilai 0 maka program tersebut akan di kembalikan ke $mystery(a+a, b/2)$. Dan jika variabel B modular 2 = 0 maka akan di kembalikan pada suatu rumus $mystery(a+a, b/2) + a$;

9. Jelaskan proses rekursif untuk program dibawah ini !

```
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);  
    }  
}
```

Analisa : Merupakan suatu program dengan menggunakan fungsi rekursif mystery yang menyimpan suatu variabe a dan b yang berupa data interger. Dan jika ke 2 data tersebut sama maka akan di proses menggunakan rumus yang sudah di simpan pada setiap fungsi mystery.

10. 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);
```

Analisa : Sebuah potongan kode program dimana variabel F menyimpan fungsi dari variabel n. Jika variabel n = 0 maka akan mengembalikan nilai 0, jika variabel n bernilai 1 maka akan mengembalikan nilai 1 begitupun dengan 2. Dimanapun nanti hasilnya akan di proses menggunakan rumus $2*f(n-2) + f(n-3)$.

11. Jelaskan proses rekursif untuk program dibawah ini !

```
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;
}
```

Analisa : Merupakan potongan program dengan menggunakan fungsi cube dan dengan menggunakan variabel square yang disimpan menggunakan tipe data integer, dimana jika n itu bernilai 0 maka variabel square akan dikembalikan pada rumus $(n-1) + 2*n - 1$. Dan jika fungsi cube yang menyimpan variabel n bernilai 1 maka fungsi cube akan dikembalikan menggunakan rumus $cube(n-1) + 3*(square(n)) - 3*n + 1$;

BAB III

KESIMPULAN

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Fungsi ini akan terus berjalan sampai kondisi berhenti terpenuhi, oleh karena itu dalam sebuah fungsi rekursif perlu terdapat 2 blok penting, yaitu blok yang menjadi titik berhenti dari sebuah proses rekursi dan blok yang memanggil dirinya sendiri.

Program C++ yang ditulis dengan mengkombinasikan fungsi baru yang tersedia didalam pustaka atau library, komputer tidak dapat membaca sehingga tidak bisa menampilkan hasil outputnya pada layar dengan kata lain program lama tidak bisa dibaca oleh komputer dengan compiler baru. Dengan memahami fungsi-fungsi dan kegunaan dari sebuah fungsi pada pemrograman C++ kita bisa menyelesaikan sebuah program sesuai dengan tujuan, dan dapat kita mengerti juga karena sifat dari pemrograman C++ adalah case sensitive yang artinya penulisan salah penulisan satu huruf atau satu angka saja di artikan berbeda oleh komputer.

DAFTAR PUSTAKA

Br Sidauruk, T. A. (2014). REKRUSIF (FAKTORIAL, FIBONACCY, HANOI).

OKTYAWAN RIRIN. (2011, April). PRAKTIKUM ALGORITMA STRUKTUR

DATAMODUL 1 – FUNGSI REKURSIF. 13.