Nama : Ifham Syafwan Fikri

Github : https://github.com/FikriSyafwan/Penugasan-Praktikum-Pemrograman

Nomor Mahasiswa : 24/545184/PA/23161

Kelas : KOM B

Dosen Pengampu : Muhammad Husni Santriaji

LAPORAN PRAKTIKUM

PERTEMUAN 8

Latihan

Number 1: Membuat bilangan Fibonacci ke-n dengan prinsip rekursi

Pada permasalahan pertama, soal memberikan instruksi untuk menyelesaikan program berupa angka ke-n dari deret Fibonacci dengan menggunakan fungsi rekursi. Angka ke-n yang dimaksud adalah Un pada dari suatu deret aritmatika

U1 + U2 + U3 +…+ Un

Dengan memberikan program nilai "n", program akan mengeluarkan nilai dari Un tersebut. Berikut berupa fungsi rekursi untuk menyelesaikan permasalahan

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

int fibonacci(int n) {
    if (n <= 1) {
        return n;
    }
    else {
        return fibonacci(n-1) + fibonacci(n-2);
    }
}
```

Langkah pertama dalam memakai fungsi rekursi adalah deklarasi fungsi itu sendiri. Fungsi rekursi yang dipakai akan diberikan nama "Fibonacci" dengan satu parameter berupa integer n untuk memasukkan nilai ke fungsi dan mengeluarkan hasil yang diinginkan. Pada baris pertama, terdapat struktur *if/else* dengan if (n <= 1) merupakan pembatas rekursif, yaitu apabila n menyentuh nilai kurang dari 1, rekursif akan mengembalikan nilai n itu sendiri.

Kemudian, else akan mengembalikan nilai berupa fungsi yang merupakan persyaratan agar fungsi rekursif dapat berjalan. Berikut suatu persamaan sebagai representasi fungsi, misalkan n adalah 3

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

fibonacci(2) + fibonacci(1)

((fibonacci(2-1) + fibonacci(0)) + 1)

((1 + 0) + 1)

2

Apabila ditulis sebagai deret, persamaan yang didapatkan adalah

1 + 1 + 2 + … + Un

Dapat dilihat bahwa hasil dari persamaan rekursif sama dengan persamaan deret.

```
int main() {
    int n = 5;
    vector<int> DeretFib;

    for (int i = 1; i <= n; i++) {
        if (i <= 2) {
        DeretFib.push_back((1));
        }
        else {
            DeretFib.push_back(DeretFib[i-2] + DeretFib[i-3]);
        }
    }

    cout << "Deret Fibonacci: ";
    for (int k = 0; k < n; k++) {
        if (k != n - 1) {
        cout << DeretFib[k] << " + ";
        }
        else {
            cout << "((" << DeretFib[k] << "))" << " + ...";
        }
    }

    cout << endl;

    cout << "Angka ke-" << n << " dari deret angka fibonacci adalah "
    << fibonacci(n) << endl;
```

**vector digunakan untuk memudahkan dalam mencetak deret Fibonacci dan memastikan program berjalan lancar*

Untuk membuktikan bahwa program akan berjalan lancar dan benar, program memerlukan input sementara dan hipotesis (*expected output*). Input akan diberikan nilai berupa 5 dan *expected output* berupa 5. Apabila sesuai dengan *expected output*, program akan menghasilkan keluaran "[Test succeed!! :>]", sedangkan sebaliknya program akan menghasilkan keluaran "[Test failed?! :<]."

Setelah menjalankan program, program akan menghasilkan hasil sebagai berikut

```
./number_1_test
Deret Fibonacci: 1 + 1 + 2 + 3 + ((5)) + ...
Angka ke-5 dari deret angka fibonacci adalah 5
[Test succeed!! :>]
Testing complete!·˚ ✧ ˚
```

Number 2: Membuat rata-rata dengan menggunakan prinsip rekursi

Pada permasalahan kedua, soal memberikan masalah, yaitu menghasilkan nilai rata-rata dari bilangan yang di-input dengan menggunakan fungsi rekursi.

```
1   #include <iostream>
2   using namespace std;
3
4   double average(double num[], int i, int n) {
5       if (i == n - 1) {
6           return num[i];
7       }
8       else {
9           return (num[i] + average(num, i+1, n));
10      }
11  }
12
13  double revAverage(double num[], int n) {
14      return average(num, 0, n) / n;
15  }
```

Langkah pertama untuk membuat fungsi rekursi yang menghasilkan nilai rata-rata adalah pendeklarasian fungsi. Deklarasi fungsi rekursi akan dilakukan dua kali dengan masing-masing fungsi diberi nama average dan revAverage. Tujuan dari dua deklarasi adalah untuk memudahkan pemanggilan fungsi dengan parameter larik dan jumlah elemen dari larik.

Pada fungsi rekursif average, baris pertama, terdapat struktur *if/ else* dengan *if* memiliki kondisi apabila i sama dengan n – 1, yaitu nilai akhir dari larik maka fungsi akan mengembalikan larik num[i]. Kemudian, baris selanjutnya, else memiliki kondisi apabila n merupakan bukan n-1, yaitu selain nilai akhir, fungsi akan mengembalikan larik num[i] ditambah fungsi rekursif average dengan parameter i + 1.

Kemudian, pada fungsi rekursif revAverage, dideklarasikan hanya dua parameter. Fungsi ini akan mengembalikan fungsi rekursif average, dengan i sama dengan nol, dibagi dengan n, yaitu jumlah elemen angka pada larik.

Dapat disimpulkan bahwa program dari fungsi rekursif yang saya gunakan akan dimulai dari i = 0, diambil dari argumen di dalam fungsi revAverage, yaitu i sebagai indeks larik pada fungsi average. Oleh karena itu, fungsi akan berjalan dari num[0] yang kemudian bertambah indeks lariknya hingga memenuhi kondisi dari *if* pada fungsi average, yaitu fungsi akan mengembalikan nilai larik num terakhir. Untuk memudahkan penjelasan, berikut penjelasan secara matematis

$$\text{Num}[n] = \{1, 2, 3, 4, 5\};\ n = 5$$

$$\text{Num}[i] + (\text{Num}[i+1] + \ldots + \text{Num}[n - 1])$$

$$\text{Num}[0] + (\text{Num}[1])$$

$$\text{Num}[0] + (\text{Num}[1] + \text{Num}[2])$$

$$\text{Num}[0] + (\text{Num}[1] + (\text{Num}[2] + \ldots + (\text{Num}[4])))$$

$$1 + (2 + (3 + \ldots + (5))) / n$$

$$15 / 5$$

$$3$$

```
17  int main() {
18      int j = 0;
19
20      int n = 5;
21      double num[n] = {1, 2, 3, 4, 5};
22
23      cout << "Maka rata-rata dari angka ";
24      for (int i = 0; i < n; i++) {
25          if (i == n - 1) {
26          cout << "dan " << num[i] << " adalah ";
27          }
28          else {
29              cout << num[i] << ", ";
30          }
31      }
32
33     for(int i = 0; i < n; i++) {
34      if (num[i] == 0) {
35          j += 1;
36      }
37     }
38
39      n = n - j;
40      cout << endl;
41      cout << revAverage(num, n) << endl;
42
```

**iterative dan deklarasi j digunakan untuk memudahkan dalam mencetak rata-rata dan memastikan program berjalan lancar*

Kemudian, pada program terdapat iterasi int j untuk menghitung nilai nol pada pendeklarasian larik. Apabila pada suatu indeks larik memiliki nilai 0, nilai j akan mengurangi nlai dari jumlah larik untuk menghasilkan nilai yang setidaknya tepat, misalnya pendeklarasian tiga angka berturut-turut 3, 0, 0 maka hasilnya adalah 3 karena 0 akan dianggap tidak ada. Namun, perlu diketahui, prinsip ini tidak berfungsi apabila pendeklarasian indeks pertamanya berupa 0.

Untuk membuktikan bahwa program akan berjalan lancar dan benar, program memerlukan input sementara dan hipotesis (*expected output*). Input akan diberikan nilai berupa jumlah elemen larik berupa 5 dan larik num[n] = {1, 2, 3, 4, 5} serta *expected output* berupa 3, yaitu rata-rata dari semua elemen bilangan pada larik. Apabila sesuai dengan *expected output*, program akan menghasilkan keluaran "[Test succeed!! :>]", sedangkan sebaliknya program akan menghasilkan keluaran "[Test failed?! :<]."

```
./number_2_test
Maka rata-rata dari angka 1, 2, 3, 4, dan 5 adalah
3
[Test succeed!! :>]
Testing complete!･ﾟ✧ﾟ
```

Pertanyaan

Question 1: Apa yang diketahui dari fungsi rekursi

Fungsi rekursif merupakan suatu metode perulangan yang menggunakan atau memanggil fungsi itu sendiri. Pemanggilan ini akan diulangi hingga suatu kondisi tercapai. Fungsi rekursif sering kali digunakan untuk menyelesaikan suatu permasalahan komputasi skala besar untuk efisiensi. Tidak hanya itu, fungsi rekursif berfungsi juga untuk merapikan dan memberikan kejelasan dalam menulis *code.*

Question 2: Jelaskan program pada nomor satu dan tuliskan persamaan yang digunakan

Langkah pertama dalam memakai fungsi rekursi adalah deklarasi fungsi itu sendiri. Fungsi rekursi yang dipakai akan diberikan nama "Fibonacci" dengan satu parameter berupa integer n untuk memasukkan nilai ke fungsi dan mengeluarkan hasil yang diinginkan. Pada baris pertama, terdapat struktur *if/else* dengan if (n <= 1) merupakan pembatas rekursif, yaitu apabila n menyentuh nilai kurang dari 1, rekursif akan mengembalikan nilai n itu sendiri. Kemudian, else akan mengembalikan nilai berupa fungsi yang merupakan persyaratan agar fungsi rekursif dapat berjalan. Berikut suatu persamaan sebagai representasi fungsi, misalkan n adalah 3

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

fibonacci(2) + fibonacci(1)

((fibonacci(2-1) + fibonacci(0)) + 1)

((1 + 0) + 1)

2

Apabila ditulis sebagai deret, persamaan yang didapatkan adalah

1 + 1 + 2 + … + Un

Question 3: Jelaskan program pada nomor dua dan tuliskan persamaan yang digunakan

Langkah pertama untuk membuat fungsi rekursi yang menghasilkan nilai rata-rata adalah pendeklarasian fungsi. Deklarasi fungsi rekursi akan dilakukan dua kali dengan masing-masing fungsi diberi nama average dan revAverage. Tujuan dari dua deklarasi adalah untuk memudahkan pemanggilan fungsi dengan parameter larik dan jumlah elemen dari larik.

Pada fungsi rekursif average, baris pertama, terdapat struktur *if/ else* dengan *if* memiliki kondisi apabila i sama dengan n – 1, yaitu nilai akhir dari larik maka fungsi akan mengembalikan larik num[i]. Kemudian, baris selanjutnya, else memiliki kondisi apabila n merupakan bukan n-1, yaitu selain nilai akhir, fungsi akan mengembalikan larik num[i] ditambah fungsi rekursif average dengan parameter i + 1.

Kemudian, pada fungsi rekursif revAverage, dideklarasikan hanya dua parameter. Fungsi ini akan mengembalikan fungsi rekursif average, dengan i sama dengan nol, dibagi dengan n, yaitu jumlah elemen angka pada larik.

Dapat disimpulkan bahwa program dari fungsi rekursif yang saya gunakan akan dimulai dari i = 0, diambil dari argumen di dalam fungsi revAverage, yaitu i sebagai indeks larik pada fungsi average. Oleh karena itu, fungsi akan berjalan dari num[0] yang kemudian bertambah indeks lariknya hingga memenuhi kondisi dari *if* pada fungsi average, yaitu fungsi akan mengembalikan nilai larik num terakhir. Untuk memudahkan penjelasan, berikut penjelasan secara matematis

$$\text{Num}[n] = \{1, 2, 3, 4, 5\};\ n = 5$$

$$\text{Num}[i] + (\text{Num}[i+1] + \ldots + \text{Num}[n - 1])$$

$$\text{Num}[0] + (\text{Num}[1])$$

$$\text{Num}[0] + (\text{Num}[1] + \text{Num}[2])$$

$$\text{Num}[0] + (\text{Num}[1] + (\text{Num}[2] + \ldots + (\text{Num}[4])))$$

$$1 + (2 + (3 + \ldots + (5))) / n$$

$$15 / 5$$

$$3$$