

LAPORAN PRAKTIKUM

“Langkah Praktikum 5: FUNGSI DAN KOMPOSISI DUA FUNGSI”

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Matematika Diskrit yang di
ampu oleh:

Nur Rochmah Dyah PA, S.T., M.Kom



Disusun Oleh:

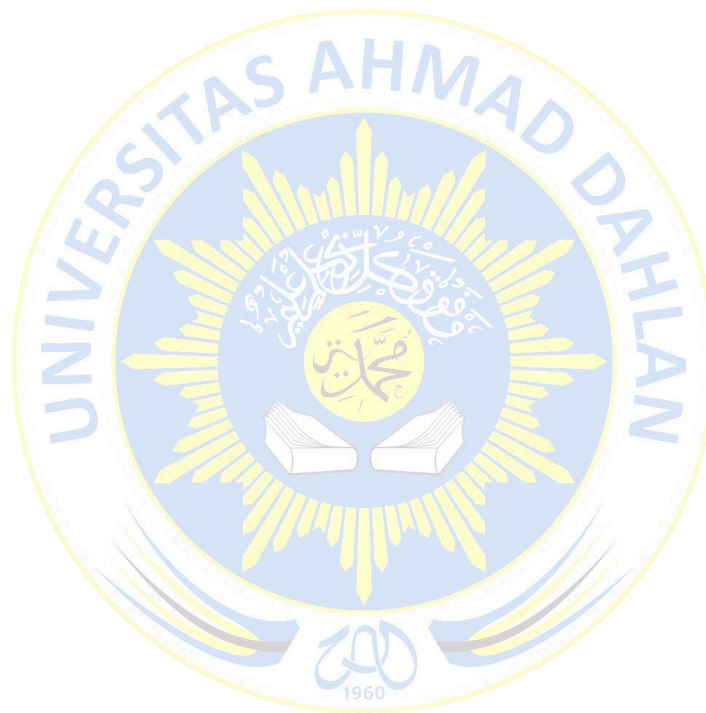
Mohammad Farid Hendianto 2200018401

Selasa 12.00-13.30

PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

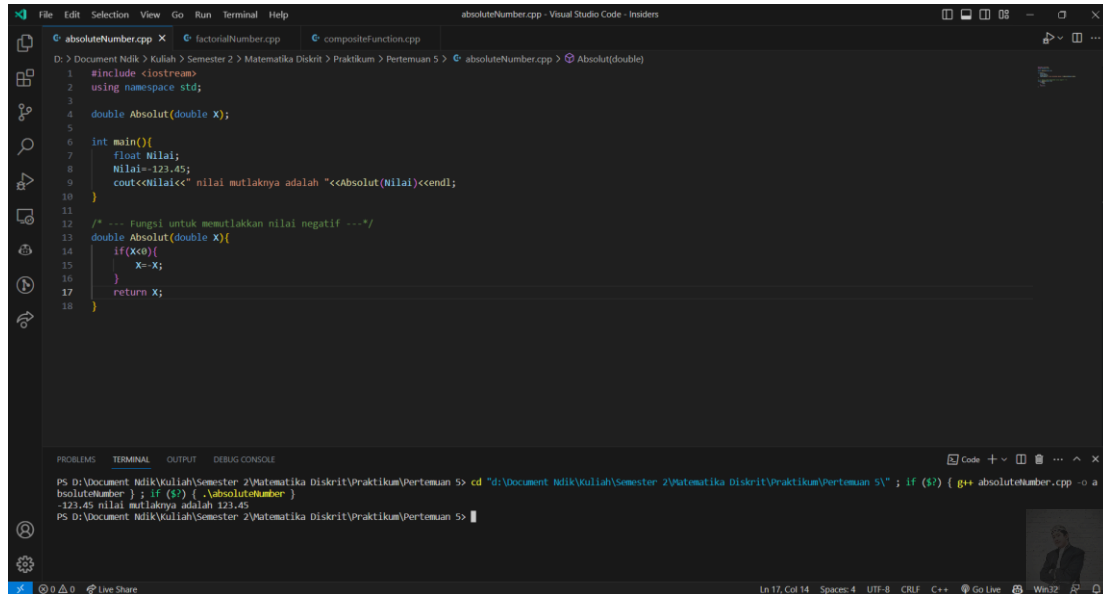
Contents

1. Diberikan program mengubah angka absolute berikut ini:	3
2. Di berikan program factorial di bawah ini :	4
3. Diberikan program komposisi dua fungsi	5



1. Diberikan program mengubah angka absolute berikut ini:

Berikut adalah kodingan beserta outputnya:



```

1 #include <iostream>
2 using namespace std;
3
4 double Absolut(double X);
5
6 int main(){
7     float Nilai;
8     Nilai=-123.45;
9     cout<<Nilai<<" nilai mutlaknya adalah "<<Absolut(Nilai)<<endl;
10 }
11
12 /* --- Fungsi untuk memulakan nilai negatif --- */
13 double Absolut(double X){
14     if(X<0){
15         X=-X;
16     }
17     return X;
18 }
  
```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

```

PS D:\Document\Hdk\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5> cd "d:\Document\Hdk\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5" ; if ($?) { g++ absoluteNumber.cpp -o absoluteNumber } ; if ($?) { .\absoluteNumber }
-123.45 nilai mutlaknya adalah 123.45
PS D:\Document\Hdk\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5>
  
```

Program di atas adalah sebuah program C++ yang memiliki satu fungsi, yaitu Absolut() yang digunakan untuk mengembalikan nilai absolut dari suatu bilangan. Berikut adalah alur programnya:

2. Program dimulai dengan mendefinisikan fungsi double Absolut(double X), dimana X merupakan parameter input dari fungsi tersebut.
3. Kemudian, program dilanjutkan dengan mendefinisikan fungsi main().
4. Pada fungsi main(), sebuah variabel float Nilai dideklarasikan dan diinisialisasi dengan nilai -123.45
5. Selanjutnya, program mencetak nilai Nilai pada layar bersamaan dengan nilai absolutnya yang dipanggil menggunakan fungsi Absolut(Nilai).
6. Fungsi Absolut() menerima parameter input X dan melakukan pengecekan. Jika nilai X kurang dari 0, maka nilai X akan dikalikan dengan -1 sehingga nilainya menjadi positif.
7. Setelah itu, fungsi Absolut() akan mengembalikan nilai X yang sudah diubah menjadi positif ke dalam fungsi main().
8. Program diakhiri dengan keluar dari fungsi main().

Dengan demikian, program ini bertujuan untuk menghitung nilai absolut dari sebuah bilangan dengan menggunakan fungsi Absolut(). Saat program dijalankan, program akan mencetak nilai absolut dari variabel Nilai yang telah diinisialisasi dengan nilai -123.45.

2. Di berikan program factorial di bawah ini :

Berikut adalah kodingan beserta outputnya:

```

1 #include <stdio.h>
2 long int Fak_Rekursif (int N);
3
4 main()
5 {
6     int N; N=5;
7     printf("5d faktorial = %ld\n", N, Fak_Rekursif(N));
8 }
9
10 long int Fak_Rekursif (int N)
11 {
12     if (N<=1)
13         return 1;
14     else
15         return N*Fak_Rekursif(N-1);
16 }

```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

```

PS D:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5> cd "d:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5" ; if ($?) { g++ factorialNumber.cpp -o factorialNumber } ; if ($?) { ./factorialNumber }
5 faktorial = 120
PS D:\Document Ndik\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5>

```

Program di atas adalah sebuah program dalam bahasa C yang menghitung nilai faktorial dari suatu bilangan menggunakan rekursi. Berikut adalah alur programnya:

1. Program dimulai dengan mendefinisikan fungsi Fak_Rekursif() yang menerima parameter input N bertipe integer dan akan mengembalikan hasil faktorial dalam bentuk long integer.
2. Kemudian, program dilanjutkan dengan mendefinisikan fungsi main().
3. Pada fungsi main(), sebuah variabel integer N dideklarasikan dan diinisialisasi dengan nilai 5.
4. Selanjutnya, program mencetak nilai variabel N pada layar disertai nilai faktorial dari N yang dipanggil menggunakan fungsi Fak_Rekursif(N).
5. Fungsi Fak_Rekursif() akan melakukan pengecekan apakah nilai N kurang dari atau sama dengan 1. Jika ya, maka fungsi akan mengembalikan hasil 1 sebagai nilai faktorial dari N.
6. Jika tidak, fungsi akan memanggil dirinya sendiri (rekursi) dengan mengirimkan parameter N-1 dan mengalikan nilai N dengan nilai kembalian fungsi tersebut (hasil faktorial dari N-1).
7. Proses rekursi akan berhenti jika nilai N sama dengan 1, dan hasil akhir dari fungsi Fak_Rekursif() akan dikembalikan ke panggilan fungsi sebelumnya atau ke fungsi main().
8. Program diakhiri dengan keluar dari fungsi main().

Dengan demikian, program ini bertujuan untuk menghitung nilai faktorial dari suatu bilangan menggunakan rekursi. Saat program dijalankan, program akan mencetak nilai faktorial dari

variabel N yang telah diinisialisasi dengan nilai 5. Hasilnya adalah 120, yang merupakan hasil dari operasi $5!$ (5 faktorial).

3. Diberikan program komposisi dua fungsi

Berikut adalah kodingan beserta outputnya:

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  // Run this program using the console window or add your own getch, system("pause") or input loop!
6
7  int main(int argc, char *argv[]) {
8      int j;
9
10     cout<<endl;
11     cout<<endl;
12     cout<<endl;
13     cout<<endl;
14     cout<<endl;
15     cout<<endl;
16     cout<<endl;
17     cout<<endl;
18     cout<<endl;
19     cout<<endl;
20     cout<<endl;
21     cout<<endl;
22     cout<<endl;
23     cout<<endl;
24     cout<<endl;
25     cout<<endl;
26     cout<<endl;
27     cout<<endl;
28     cout<<endl;
29     cout<<endl;
30     cout<<endl;
31     cout<<endl;
32     cout<<endl;
33     cout<<endl;
34     cout<<endl;
35     cout<<endl;
36     cout<<endl;
37     cout<<endl;
38     cout<<endl;
39     cout<<endl;
40     cout<<endl;
41     cout<<endl;
42     cout<<endl;
43     cout<<endl;
44     cout<<endl;
45     cout<<endl;
46     cout<<endl;
47     cout<<endl;
48     cout<<endl;
49     cout<<endl;
50     cout<<endl;
51     cout<<endl;
52     cout<<endl;
53     cout<<endl;
54     cout<<endl;
55     cout<<endl;
56     cout<<endl;
57     cout<<endl;
58     cout<<endl;
59     cout<<endl;
60     cout<<endl;
61     cout<<endl;
62     cout<<endl;
63     cout<<endl;
64     cout<<endl;
65     cout<<endl;
66     cout<<endl;
67     cout<<endl;
68     cout<<endl;
69     cout<<endl;
70     cout<<endl;
71     cout<<endl;
72     cout<<endl;
73     cout<<endl;
74     cout<<endl;
75     cout<<endl;
76     cout<<endl;
77     cout<<endl;
78     cout<<endl;
79     cout<<endl;
80     cout<<endl;
81     cout<<endl;
82     cout<<endl;
83     cout<<endl;
84     cout<<endl;
85     cout<<endl;
86     cout<<endl;
87     cout<<endl;
88     cout<<endl;
89     cout<<endl;
90     cout<<endl;
91     cout<<endl;
92     cout<<endl;
93     cout<<endl;
94     cout<<endl;
95     cout<<endl;
96     cout<<endl;
97     cout<<endl;
98     cout<<endl;
99     cout<<endl;
100    cout<<endl;
101    return 0;
102 }
    
```

Berikut output program ke tiga

```

PS D:\Document Ndidk\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5> cd "d:\Document Ndidk\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5" ; if ($?) { g++ compositefunction.cpp -o compositefunction } ; if ($?) { .\compositefunction }
Masukkan jumlah f(x) 12
Masukkan fungsi f->x
f(x):5
x:4
f(x):5
x:4
Masukkan fungsi g->x
g(x):5
x:5
g(x):4
x:5
f(x) = {(5)(4),(5)(4),}
g(x) = {(5)(5),(4)(5),}
fog(x)={(5,5),(5,5),}
PS D:\Document Ndidk\Kuliah\Semester 2\Matematika Diskrit\Praktikum\Pertemuan 5>
    
```

Program di atas adalah sebuah program C++ yang berfungsi untuk menghitung komposisi dari dua fungsi $f(x)$ dan $g(x)$, yaitu $fog(x)$. Berikut adalah alur programnya:

Program dimulai dengan mendeklarasikan variabel integer jumlah, serta array of string $f[100][100]$ dan $g[100][100]$.

Kemudian, program menanyakan kepada pengguna berapa jumlah fungsi yang ingin dimasukkan.

Selanjutnya, program meminta pengguna memasukkan masing-masing fungsi $f(x)$ beserta nilai x -nya, kemudian menyimpannya ke dalam array $f[][]$.

Setelah itu, program meminta pengguna memasukkan masing-masing fungsi $g(x)$ beserta nilai x -nya, kemudian menyimpannya ke dalam array $g[][]$.

Program mencetak fungsi $f(x)$ dan $g(x)$ yang telah dimasukkan oleh pengguna dengan format: "(fungsinya)(nilainya)", kemudian menambahkan kurung kurawal sebagai penanda akhir dari setiap fungsi tersebut. Contohnya: $f(x) = \{(x+1)(2), (2x)(3), \}$, $g(x) = \{(2x-1)(1), (3x+1)(0), \}$.

Program kemudian melakukan perhitungan komposisi fungsi $f(x)$ dan $g(x)$ dengan mencari pasangan nilai x pada kedua fungsi yang sama. Jika ditemukan pasangan nilai x yang sama antara fungsi f dan g , maka program akan mencetak pasangan nilai tersebut dalam bentuk $(f, \text{nilai_fog})$.

Setelah selesai melakukan pencarian, program mencetak hasil komposisi fungsi $fog(x)$ dengan format yang sama seperti pada langkah 5.

Program diakhiri dengan keluar dari fungsi $\text{main}()$.

Contoh output program:

Masukkan jumlah $f(x)$:2

Masukkan fungsi $f \rightarrow x$

$f(x):(x+1)$

$x:2$

$f(x):(2x)$

$x:3$

Masukkan fungsi $g \rightarrow x$

$g(x):(2x-1)$

$x:1$

$$g(x) : (3x+1)$$

$$x : \emptyset$$

$$f(x) = \{(x+1)(2), (2x)(3), \}$$

$$g(x) = \{(2x-1)(1), (3x+1)(\emptyset), \}$$

$$f \circ g(x) = \{(2x, 1), \}$$

Output tersebut menunjukkan bahwa ketika fungsi $f(x)$ dan $g(x)$ adalah $(x+1)(2)$ dan $(2x-1)(1)$ secara berturut-turut, maka hasil komposisi $f \circ g(x)$ nya adalah $(2x, 1)$. Artinya, komposisi fungsi $f(x)$ dengan $g(x)$ pada nilai $x=1$ menghasilkan nilai $2x$. Sedangkan pada contoh input lainnya, tidak ditemukan pasangan nilai x yang sama antara fungsi $f(x)$ dan $g(x)$, sehingga hasil komposisi $f \circ g(x)$ nya kosong.

