

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
CODE BLOCKS IDE &
PENGENALAN BAHASA C++**



Disusun oleh :

Junadil Muqorobin (103112400281)

Dosen

Fahrudin Mukti Wibowo S.Kom., M.Eng

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Pemrograman dengan bahasa C++ merupakan salah satu fondasi penting dalam pembelajaran struktur data. C++ termasuk bahasa pemrograman tingkat tinggi yang mendukung paradigma prosedural maupun berorientasi objek.

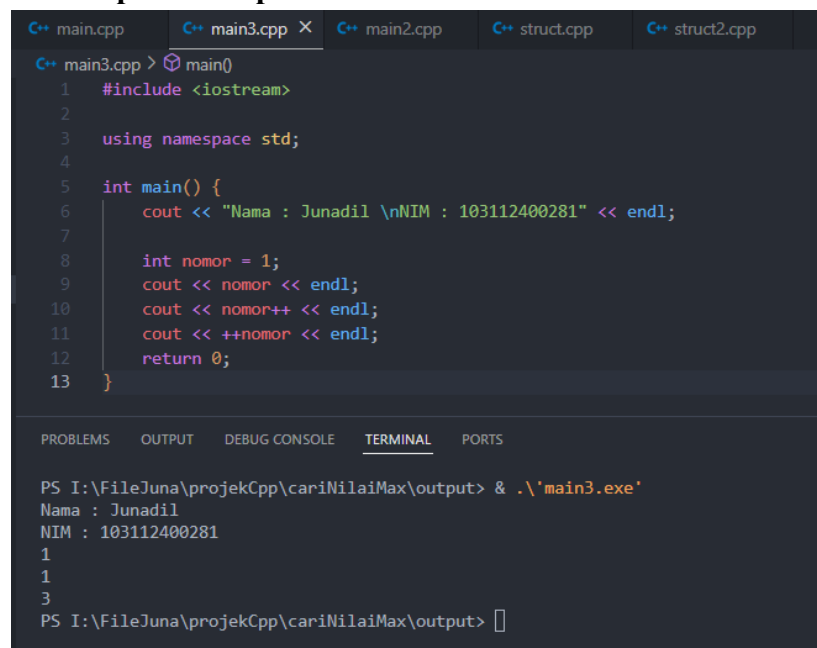
Struktur dasar program C++ diawali dengan direktif `#include` yang berfungsi menyertakan library standar, misalnya `<iostream>` untuk operasi input dan output. Program utama selalu diawali oleh fungsi `main()`, tempat semua perintah dieksekusi. Untuk interaksi dengan pengguna, C++ menyediakan objek `cin` (input) dan `cout` (output) yang memanfaatkan operator ekstraksi (`>>`) dan inserter (`<<`). Dengan fasilitas ini, pengguna dapat memberikan masukan data, sedangkan program menampilkan hasil pengolahan data tersebut. Konsep operator sangat penting dalam pemrograman. Operator aritmatika seperti `+`, `-`, `*`, dan `/` digunakan untuk operasi matematis, sedangkan operator increment dan decrement (`++` dan `--`) digunakan untuk menambah atau mengurangi nilai variabel secara cepat. Pemahaman perbedaan *post-increment* (`x++`) dan *pre-increment* (`++x`) juga menjadi dasar agar mahasiswa tidak keliru dalam menuliskan logika program. Struktur kendali dalam C++ meliputi percabangan dan perulangan. Percabangan dengan `if`, `if-else`, maupun `switch` digunakan untuk pengambilan keputusan berdasarkan kondisi tertentu. Perulangan dengan `for`, `while`, dan `do..while` digunakan untuk mengeksekusi perintah berulang kali sesuai syarat yang diberikan. Perulangan ini penting untuk mengurangi pengulangan kode secara manual dan membuat program lebih efisien.

Selain tipe data dasar (int, float, char, string), C++ juga menyediakan tipe data bentukan seperti struct yang memungkinkan pengelompokan beberapa atribut ke dalam satu entitas. Misalnya, data mahasiswa dapat terdiri dari nama dan NIM yang dikemas dalam satu struktur. Hal ini mempermudah pengelolaan data yang lebih kompleks dan terorganisasi.

Berdasarkan dasar teori tersebut, praktikum ini berfokus pada pengenalan elemen-elemen dasar C++ melalui latihan *guided* dan *unguided*. Materi meliputi penggunaan input-output, percabangan, perulangan, operator, serta tipe data bentukan. Pemahaman terhadap teori ini menjadi langkah awal untuk mempelajari konsep struktur data yang lebih kompleks pada pertemuan berikutnya.

B. Guided

1. Menampilkan output dan increment



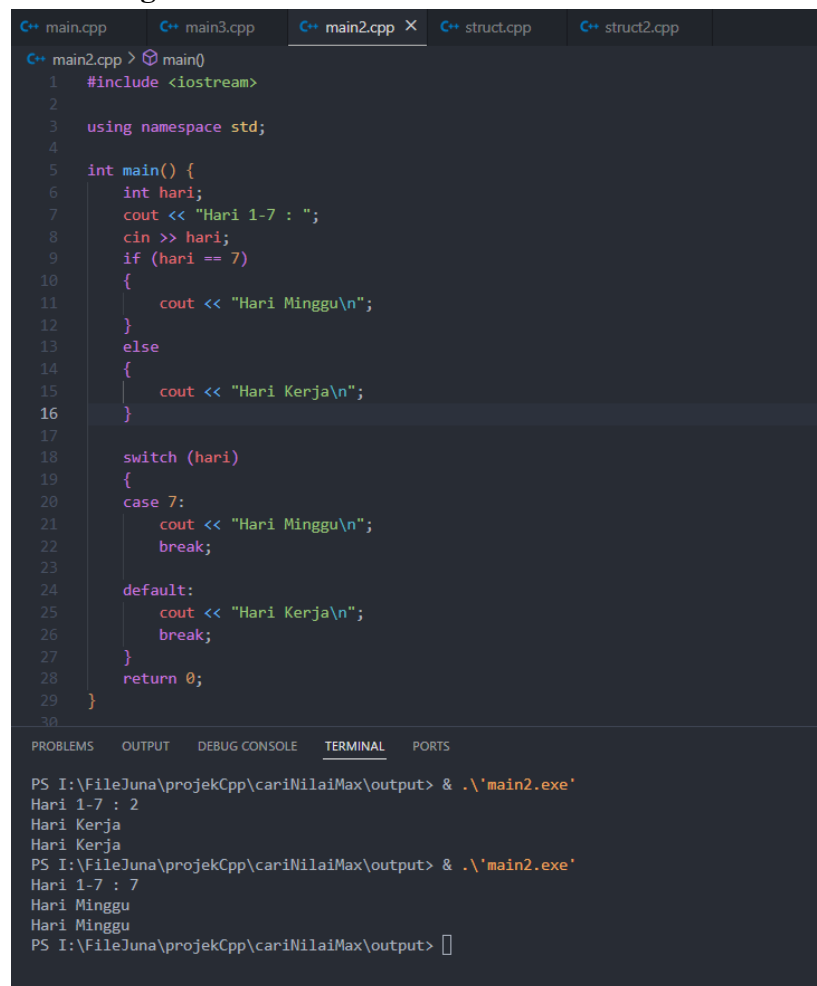
```
C++ main.cpp | C++ main3.cpp X | C++ main2.cpp | C++ struct.cpp | C++ struct2.cpp
C++ main3.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      cout << "Nama : Junadil \nNIM : 103112400281" << endl;
7
8      int nomor = 1;
9      cout << nomor << endl;
10     cout << nomor++ << endl;
11     cout << ++nomor << endl;
12     return 0;
13 }
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
PS I:\FileJuna\projekCpp\cariNilaiMax\output> & .\main3.exe
Nama : Junadil
NIM : 103112400281
1
1
3
PS I:\FileJuna\projekCpp\cariNilaiMax\output> 
```

Source code diatas digunakan untuk menampilkan output dan melihat perbedaan antara beberapa gaya increment, jika nomor++ berarti tampilkan dulu baru ditambah 1, sedangkan ++nomor berarti tambah 1 baru ditampilkan atau sering disebut pre-increment dan post-increment.

2. Percabangan



```
C++ main.cpp C++ main3.cpp C++ main2.cpp X C++ struct.cpp C++ struct2.cpp
C++ main2.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int hari;
7      cout << "Hari 1-7 : ";
8      cin >> hari;
9      if (hari == 7)
10     {
11         cout << "Hari Minggu\n";
12     }
13     else
14     {
15         cout << "Hari Kerja\n";
16     }
17
18     switch (hari)
19     {
20     case 7:
21         cout << "Hari Minggu\n";
22         break;
23
24     default:
25         cout << "Hari Kerja\n";
26         break;
27     }
28     return 0;
29 }
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS I:\FileJuna\projekCpp\cariNilaiMax\output> & .\'main2.exe'
Hari 1-7 : 2
Hari Kerja
Hari Kerja
PS I:\FileJuna\projekCpp\cariNilaiMax\output> & .\'main2.exe'
Hari 1-7 : 7
Hari Minggu
Hari Minggu
PS I:\FileJuna\projekCpp\cariNilaiMax\output> 
```

Program tersebut digunakan untuk memeriksa apakah hari ke-n yang diinputkan user merupakan hari kerja

atau hari minggu. Pertama input user akan masuk ke variable hari menggunakan cin. Lalu di periksa menggunakan if, jika input angka 7 maka output hari minggu, dan input selain itu adalah hari kerja.

Selain itu juga bisa menggunakan switch dengan case 7 maka akan menghasilkan output hari minggu, dan setting default digunakan untuk output selain 7 sebagai hari kerja.

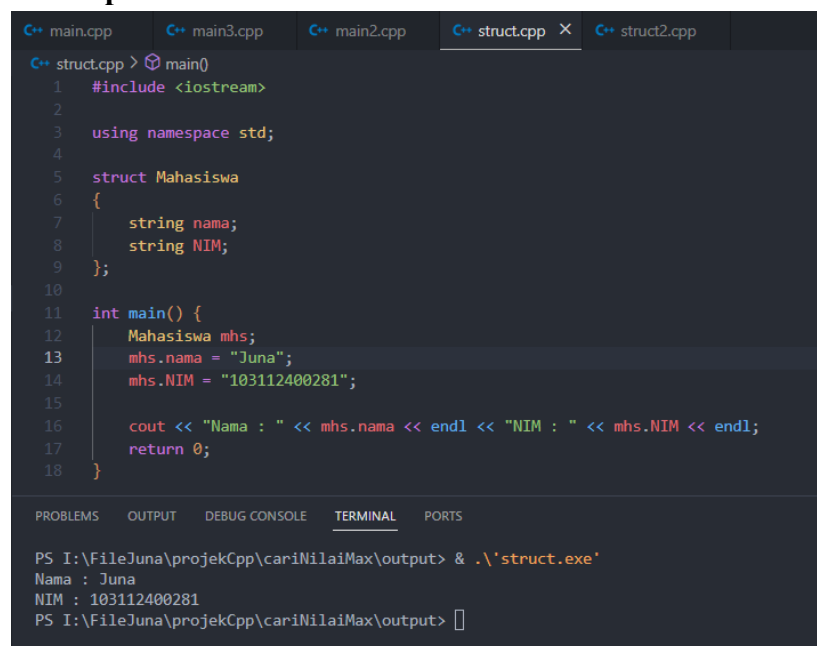
3. Perulangan

```
C++ main.cpp X C++ main3.cpp C++ main2.cpp C++ struct.cpp C++ struct2.cpp
C++ main.cpp > main()
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     for (int i = 0; i < 10; i++)
8     {
9         cout << "Juna\n";
10    }
11    cout << endl;
12
13    int i = 0;
14    while (i < 10)
15    {
16        cout << "103112400281\n";
17        i++;
18    }
19    cout << endl;
20
21    int j = 0;
22    do
23    {
24        cout << j;
25    } while (j != 0);
26
27    return 0;
28
29
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS I:\FileJuna\projekCpp\cariNilaiMax\output> cd 'I:\FileJuna\projekCpp\cariNilaiMax\output'
PS I:\FileJuna\projekCpp\cariNilaiMax\output> & .\main.exe
Juna
Juna
Juna
Juna
Juna
Juna
Juna
Juna
Juna
Juna
103112400281
103112400281
103112400281
103112400281
103112400281
103112400281
103112400281
103112400281
103112400281
103112400281
0
0F I:\FileJuna\projekCpp\cariNilaiMax\output>
```

Program ini menunjukkan tiga jenis perulangan yang ada di cpp yaitu for, while, dan do while. Pada for kita perlu mendefinisikan nilai awal terlebih dahulu yaitu I dimulai dari 0, kemudian jika $i < 10$ maka program akan terus berjalan, dan setiap iterasi nilai i akan ditambah 1, dan jika kondisi terpenuhi maka akan menjalankan perintahnya yaitu output nama.

Sama halnya dengan for, while juga harus mendefinisikan nilai i dan kondisi, namun increment dilakukan setelah perintah atau output dijalankan. Untuk do while, kita harus definisikan juga j di awal, lalu akan langsung menjalankan perintah, dan setelah itu baru menjalankan kondisi, jadi dengan do while pasti ada satu perintah atau output yang dijalankan.

4. Struct pertama



```
C++ main.cpp  C++ main3.cpp  C++ main2.cpp  C++ struct.cpp X  C++ struct2.cpp

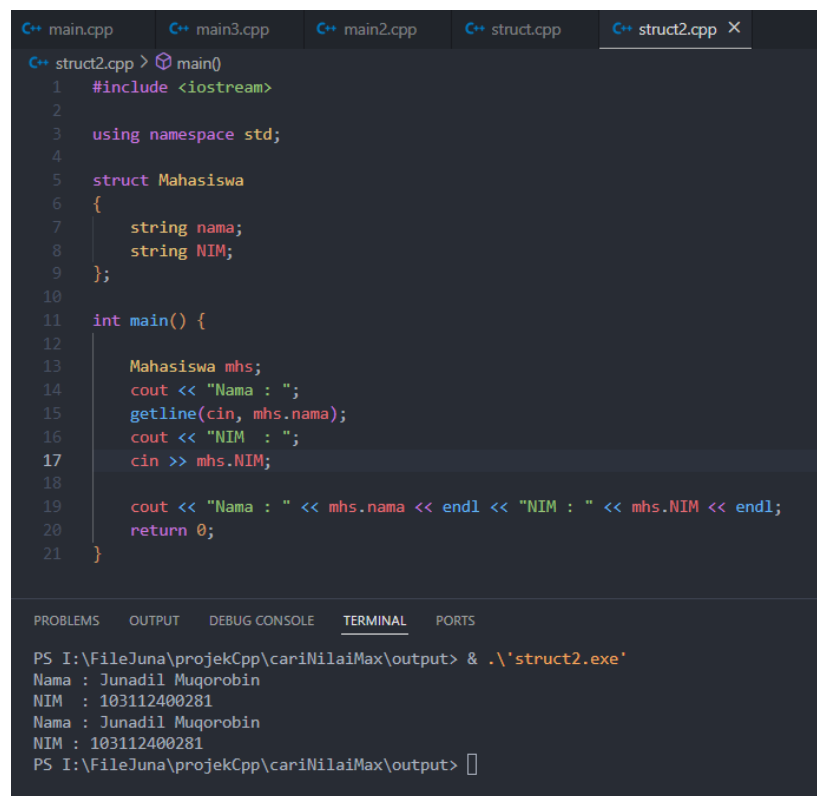
C++ struct.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  struct Mahasiswa
6  {
7      string nama;
8      string NIM;
9  };
10
11 int main() {
12     Mahasiswa mhs;
13     mhs.nama = "Juna";
14     mhs.NIM = "103112400281";
15
16     cout << "Nama : " << mhs.nama << endl << "NIM : " << mhs.NIM << endl;
17     return 0;
18 }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS I:\FileJuna\projekCpp\cariNilaiMax\output> & .\'struct.exe'
Nama : Juna
NIM : 103112400281
PS I:\FileJuna\projekCpp\cariNilaiMax\output> 
```

Program ini bertujuan untuk mengenalkan penggunaan struct, yaitu tipe data bentukan berisi Kumpulan variable. Terlihat pada struct Mahasiswa terdapat variable nama dan NIM dengan tipe data string. Variable tersebut dapat digunakan untuk menyimpan input pada function main() tanpa harus membuat variable lagi dan cara memanggilnya yaitu dengan mendeklarasikan structnya pada main() Mahasiswa mhs.

5. Struct kedua



```
C++ main.cpp C++ main3.cpp C++ main2.cpp C++ struct.cpp C++ struct2.cpp X
C++ struct2.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  struct Mahasiswa
6  {
7      string nama;
8      string NIM;
9  };
10
11 int main() {
12
13     Mahasiswa mhs;
14     cout << "Nama : ";
15     getline(cin, mhs.nama);
16     cout << "NIM : ";
17     cin >> mhs.NIM;
18
19     cout << "Nama : " << mhs.nama << endl << "NIM : " << mhs.NIM << endl;
20     return 0;
21 }
```

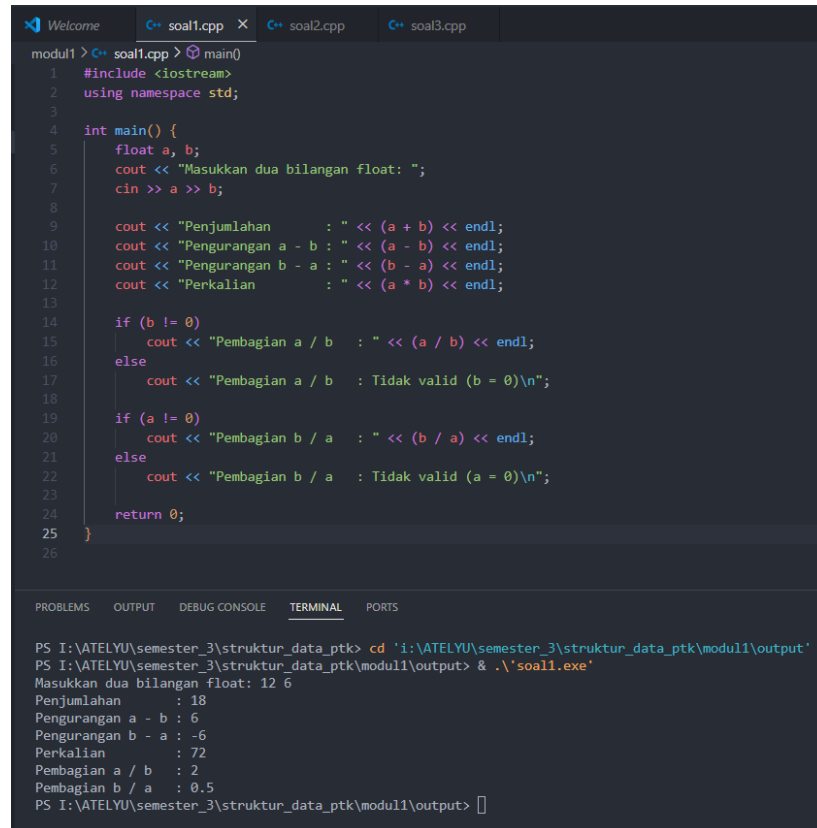
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS I:\FileJuna\projekCpp\cariNilaiMax\output> & .\'struct2.exe'
Nama : Junadil Muqorobin
NIM : 103112400281
Nama : Junadil Muqorobin
NIM : 103112400281
PS I:\FileJuna\projekCpp\cariNilaiMax\output>
```

Program tersebut mengimplementasikan input dengan menggunakan struct. Untuk dapat menulis nama dengan spasi saya menggunakan getline().

C. Unguided

1. Operasi bilangan float



```
modul1 > C++ soal1.cpp X C++ soal2.cpp C++ soal3.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     float a, b;
6     cout << "Masukkan dua bilangan float: ";
7     cin >> a >> b;
8
9     cout << "Penjumlahan      : " << (a + b) << endl;
10    cout << "Pengurangan a - b : " << (a - b) << endl;
11    cout << "Pengurangan b - a : " << (b - a) << endl;
12    cout << "Perkalian        : " << (a * b) << endl;
13
14    if (b != 0)
15        cout << "Pembagian a / b : " << (a / b) << endl;
16    else
17        cout << "Pembagian a / b : Tidak valid (b = 0)\n";
18
19    if (a != 0)
20        cout << "Pembagian b / a : " << (b / a) << endl;
21    else
22        cout << "Pembagian b / a : Tidak valid (a = 0)\n";
23
24    return 0;
25 }
26
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS I:\ATELYU\semester_3\struktur_data_ptk> cd 'i:\ATELYU\semester_3\struktur_data_ptk\modul1\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\modul1\output> & .\soal1.exe
Masukkan dua bilangan float: 12 6
Penjumlahan      : 18
Pengurangan a - b : 6
Pengurangan b - a : -6
Perkalian        : 72
Pembagian a / b   : 2
Pembagian b / a   : 0.5
PS I:\ATELYU\semester_3\struktur_data_ptk\modul1\output>
```

Program tersebut digunakan untuk menghitung operasi penjumlahan, pengurangan, pembagian, dan perkalian dari dua buah bilangan float. Pertama tama program membaca input dari user menggunakan cin yang kemudian masuk ke variable a dan b.

Untuk penjumlahan, pengurangan, dan perkalian dilakukan dengan rumus biasa. Sedangkan untuk pmbagian saya menggunakan if else agar tidak terjadi error, karena pembagi tidak boleh nol. Jika pembagi bukan nol, maka akan menjalankan rumus pembagian,

namun jika pembagi adalah 0 maka akan menghasilkan output tidak valid.

2. Konversi angka ke tulisan

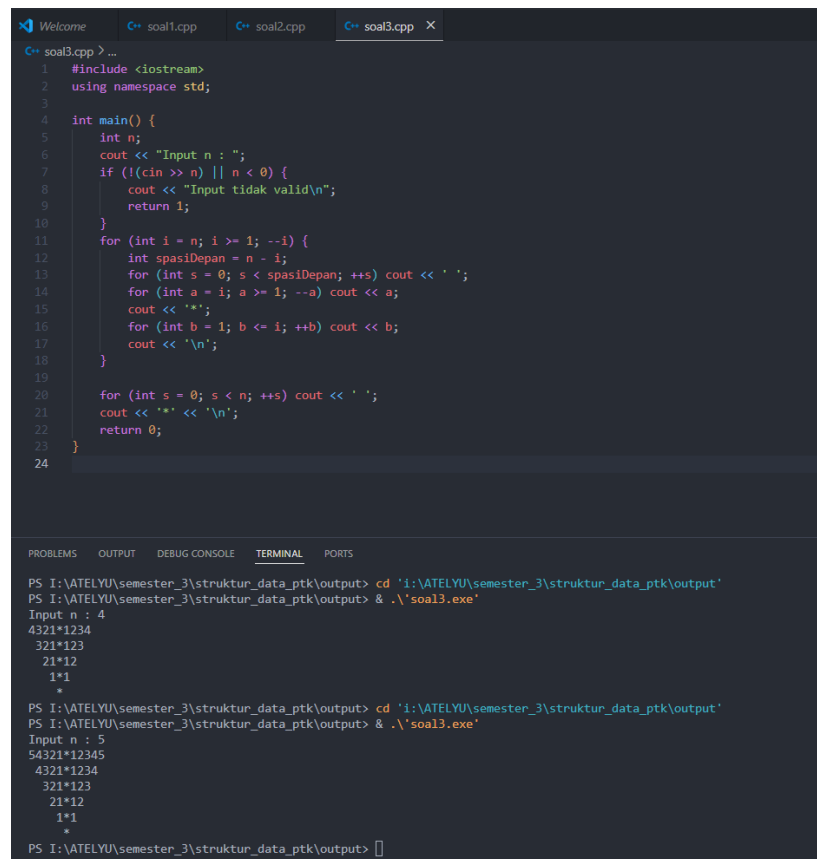


```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 string satuan(int n) {
6     switch(n) {
7         case 0: return "nol";
8         case 1: return "satu";
9         case 2: return "dua";
10        case 3: return "tiga";
11        case 4: return "empat";
12        case 5: return "lima";
13        case 6: return "enam";
14        case 7: return "tujuh";
15        case 8: return "delapan";
16        case 9: return "sembilan";
17    }
18    return "";
19 }
20
21 string konversi(int n) {
22     if (n < 0 || n > 100) return "Input di luar 0-100";
23     if (n == 100) return "seratus";
24     if (n < 10) return satuan(n);
25     if (n == 10) return "sepuluh";
26     if (n == 11) return "sebelas";
27     if (n < 20) {
28         return satuan(n % 10) + " belas";
29     }
30     int puluhan = n / 10;
31     int sisa = n % 10;
32     string hasil = satuan(puluhan) + " puluh";
33     if (sisa != 0) hasil += " " + satuan(sisa);
34     return hasil;
35 }
36
37 int main() {
38     int x;
39     cout << "Input bilangan bulat 0-100 : ";
40     if (!(cin >> x)) {
41         cout << "Input tidak valid\n";
42         return 1;
43     }
44     cout << x << " : " << konversi(x) << '\n';
45     return 0;
46 }
47
```

```
PS I:\ATELYU\semester_3\struktur_data_ptk\output> cd 'i:\ATELYU\semester_3\struktur_data_ptk\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\output> & .\'soal2.exe'
Input bilangan bulat 0-100 : 67
67 : enam puluh tujuh
PS I:\ATELYU\semester_3\struktur_data_ptk\output> & .\'soal2.exe'
Input bilangan bulat 0-100 : 9
9 : sembilan
PS I:\ATELYU\semester_3\struktur_data_ptk\output> & .\'soal2.exe'
Input bilangan bulat 0-100 : 99
99 : sembilan puluh sembilan
PS I:\ATELYU\semester_3\struktur_data_ptk\output> 
```

Program tersebut digunakan untuk mengubah bilangan bulat 0-100 menjadi text. Struct satuan() digunakan untuk menampung variable satuan dimana di dalamnya terdapat switch case untuk menentukan bilangan dan text yang sesuai. Kemudian struct konversi() berfungsi untuk mengatur aturan angka seperti belasan, puluhan, atau seratus, Misalnya input 67 : enam puluh tujuh, dan 9 : Sembilan.

3. Gambar mirror



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cout << "Input n : ";
7     if (!(cin >> n) || n < 0) {
8         cout << "Input tidak valid\n";
9         return 1;
10    }
11    for (int i = n; i >= 1; --i) {
12        int spasiDepan = n - i;
13        for (int s = 0; s < spasiDepan; ++s) cout << ' ';
14        for (int a = i; a >= 1; --a) cout << a;
15        cout << '*';
16        for (int b = 1; b <= i; ++b) cout << b;
17        cout << '\n';
18    }
19
20    for (int s = 0; s < n; ++s) cout << ' ';
21    cout << '*' << '\n';
22    return 0;
23 }
24
```

```
PS I:\ATELYU\semester_3\struktur_data_ptk\output> cd 'i:\ATELYU\semester_3\struktur_data_ptk\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\output> & .\soal3.exe
Input n : 4
4321*1234
321*123
21*12
1*1
*
PS I:\ATELYU\semester_3\struktur_data_ptk\output> cd 'i:\ATELYU\semester_3\struktur_data_ptk\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\output> & .\soal3.exe
Input n : 5
54321*12345
4321*1234
321*123
21*12
1*1
*
PS I:\ATELYU\semester_3\struktur_data_ptk\output> 
```

Program tersebut bertujuan untuk mencetak pola mirror atau simetris dari bilangan dan karakter Bintang. Pertama user melakukan input n yaitu tinggi

pola yang akan dicetak, misal $n = 5$. Kemudian for pertama digunakan untuk mengatur jumlah baris yang dimulai dari baris $i = n$, lalu berkurang sampai $i = 1$.

Int spasiDepan() digunakan agar pola rata kanan kiri, semakin kebawah maka spasi akan bertambah sehingga pola dapat simetris.

Setelah semua baris selesai, program menambahkan satu baris paling bawah dengan karakter Bintang di tengah, dengan cara cetak spasi sebanyak n lalu cetak Bintang agar posisi di tengah. Contoh dapat dilihat pada running program.

D. Kesimpulan

Dari praktikum yang telah dilakukan, dapat disimpulkan bahwa pemahaman dasar pemrograman C++ sangat penting sebelum mempelajari materi struktur data yang lebih mendalam. Melalui latihan *guided*, saya belajar bagaimana menggunakan perulangan, percabangan, operator, serta tipe data dasar dan struct secara bertahap. Sedangkan pada latihan *unguided*, saya dapat menggabungkan konsep-konsep tersebut ke dalam kasus nyata, seperti membuat operasi sederhana, mengonversi angka ke bentuk tulisan, dan mencetak pola mirror dengan perulangan bersarang. Praktikum ini menunjukkan bahwa setiap konsep memiliki perannya masing-masing. Perulangan membantu mengurangi pengulangan kode, percabangan memungkinkan program mengambil keputusan, operator digunakan dalam berbagai operasi logika maupun

matematika, dan struct mempermudah pengelompokan data yang berhubungan.

Secara keseluruhan, praktikum ini menjadi dasar yang kokoh untuk melangkah ke materi berikutnya. Pemahaman input-output, percabangan, perulangan, operator, dan struktur data sederhana akan sangat membantu ketika mempelajari algoritma serta implementasi struktur data yang lebih rumit di pertemuan selanjutnya.

E. Referensi

Kaswar, A. B., Nurjannah., & Zain, S. G. (2021). *Mudah Belajar Pemrograman Dasar C++*. Syiah Kuala University Press.

Rista, M. A., & Sujatmiko, B. (2024). *Pemanfaatan Compiler C++ Online Dalam Pengembangan Aplikasi Berbasis Web Untuk Mengukur Pengaruh Pembelajaran Discovery Learning Terhadap Kompetensi Pemrograman C++ Pada Mata Pelajaran Pemrograman Dasar*. *IT-Edu : Jurnal Information Technology and Education*, 9(2), 18–27. <https://doi.org/10.26740/it-edu.v9i2.61339>

