

Tugas Pendahuluan Modul 1 STRUKTUR
DATA - Ganjil 2024/2025
"Pengenalan C++: Subprogram & Array"

A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TESASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN = PENGURANGAN POINJURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 2 adalah Senin, 30 September 2024 pukul 07.30 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAUTIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. Codingan diupload di Github dan upload Laporan di Labmenggunakan format **PDF** dengan ketentuan: **TP_MOD_[XX]_NIM_NAMA.pdf**

CP (WA):

- Andini (082243700965)
- Imelda (082135374187)

SELAMAT MENGERJAKAN^^

**LAPORAN PRAKTIKUM
MODUL 1
“ Pengenalan C++: Subprogram & Array ”**



**Disusun Oleh:
Shilfi Habibah - 2311104002
S1SE07-01**

**Dosen :
Yudha Islami Sulistya**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

B. Soal Tugas Pendahuluan

1. (Input/Output) Tuliskan kode berikut dan jalankan. a) Masukkan nama lengkap anda dan nim anda. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban. b) Masukkan nama pertama anda dan nim anda. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban.

Code:

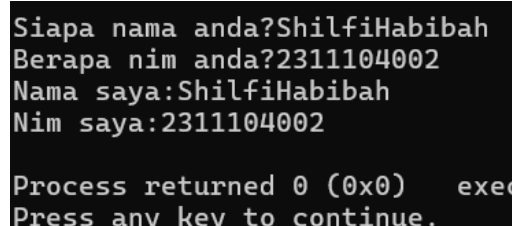
```
#include <iostream>

using namespace std;

int main()
{
    string nama, nim;
    cout << "Siapa nama anda?" ;
    cin >> nama;
    cout << "Berapa nim anda?";
    cin >> nim;
    cout << "Nama saya:" << nama << endl;
    cout << "Nim saya:" << nim << endl;
    return 0;
}
```

- **Iostream** berkas header, yang memungkinkan program untuk melakukan operasi input/output. **using namespace std;** Direktif tersebut memungkinkan program untuk menggunakan namespace standar, yang menyediakan akses ke fungsi dan objek pustaka standar.
- **Main** adalah titik masuk program, tempat eksekusi dimulai
- Dua variabel string, **nama** dan **nim**, dideklarasikan untuk menyimpan nama pengguna dan nomor ID mahasiswa, masing-masing.
- Program meminta pengguna untuk memasukkan nama dan nomor ID mahasiswa menggunakan **cout** pernyataan. **cin** Pernyataan tersebut membaca masukan pengguna dan menyimpannya dalam variabel **nama** dan **nim**.
- Program menampilkan nilai masukan kembali ke pengguna menggunakan **cout** pernyataan. Akan tetapi, ada kesalahan pada pernyataan keluaran kedua, di mana **nama** digunakan sebagai ganti **nim**. Ini harus diperbaiki menjadi **cout << "Nim saya:" << nim << endl;**.

Output:



```
Siapa nama anda?ShilfiHabibah
Berapa nim anda?2311104002
Nama saya:ShilfiHabibah
Nim saya:2311104002

Process returned 0 (0x0)   exec
Press any key to continue.
```

2. (Operasi aritmatika) Tuliskan kode berikut dan jalankan. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban.

Code:

```

#include <iostream>

using namespace std;

int main()
{
    int bil1 = 3, bil2 = 4, hasil1;
    float bil3 = 3.0, bil4 = 4.0, hasil2;
    hasil1 = bil1 + bil2;
    cout << hasil1 << endl;
    hasil1 = bil1 - bil2;
    cout << hasil1 << endl;
    hasil1 = bil1 * bil2;
    cout << hasil1 << endl;
    hasil1 = bil1 / bil2; // integer division
    cout << hasil1 << endl;
    hasil1 = bil2 / bil1; // integer division
    cout << hasil1 << endl;
    hasil1 = bil1 % bil2; // modulo
    cout << hasil1 << endl;
    hasil1 = bil2 % bil1; // modulo
    cout << hasil1 << endl;
    hasil2 = bil3 / bil4;
    cout << hasil1 << endl;
    return 0;
}

```

- **int bil1 = 3, bil2 = 4, hasil1;** : Deklarasi variabel bil1 dan bil2 dengan nilai awal 3 dan 4, serta variabel hasil1 untuk menyimpan hasil operasi.
- **float bil3 = 3.0, bil4 = 4.0, hasil2;** : Deklarasi variabel bil3 dan bil4 dengan nilai awal 3.0 dan 4.0, serta variabel hasil2 untuk menyimpan hasil operasi.
- **hasil1 = bil1 + bil2;** Operasi penjumlahan antara bil1 dan bil2, hasilnya disimpan di hasil1.
- **hasil1 = bil1 - bil2;** Operasi pengurangan antara bil1 dan bil2, hasilnya disimpan di hasil1.
- **hasil1 = bil1 * bil2;** Operasi perkalian antara bil1 dan bil2, hasilnya disimpan di hasil1.
- **hasil1 = bil1 / bil2;** Operasi pembagian antara bil1 dan bil2, hasilnya disimpan di hasil1. Perlu diingat bahwa operasi pembagian ini adalah integer division, sehingga hasilnya akan dibulatkan ke bawah.
- **hasil1 = bil1 % bil2;** Operasi modulo antara bil1 dan bil2, hasilnya disimpan di hasil1.
- **cout << hasil1 << endl;** Output hasil operasi ke layar.

Output:

```

7
-1
12
0
1
3
1
1
1

Process returned 0 (0x0)
Press any key to continue.

```

3. (Operasi perbandingan) Tuliskan kode berikut dan jalankan. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban.

Code:

```

#include <iostream>

using namespace std;

int main()
{
    int bil1 = 2, bil2 = 3, hasil;
    hasil = bil1 > bil2;
    cout << hasil << endl;
    hasil = bil1 >= bil2;
    cout << hasil << endl;
    hasil = bil1 < bil2;
    cout << hasil << endl;
    hasil = bil1 <= bil2;
    cout << hasil << endl;
    hasil = bil1 == bil2;
    cout << hasil << endl;
    hasil = bil1 != bil2;
    cout << hasil << endl;
    return 0;
}

```

- **int bil1 = 2, bil2 = 3, hasil;**: Deklarasi variabel bil1 dan bil2 dengan nilai awal 2 dan 3, serta variabel hasil untuk menyimpan hasil operasi.
- **hasil = bil1 > bil2;**: Operasi perbandingan lebih besar antara bil1 dan bil2, hasilnya disimpan di hasil. Karena bil1 (2) tidak lebih besar dari bil2 (3), maka hasilnya adalah 0 (false).
- **hasil = bil1 >= bil2;**: Operasi perbandingan lebih besar atau sama dengan antara bil1 dan bil2, hasilnya disimpan di hasil. Karena bil1 (2) tidak lebih besar atau sama dengan bil2 (3), maka hasilnya adalah 0 (false).
- **hasil = bil1 < bil2;**: Operasi perbandingan kurang dari antara bil1 dan bil2, hasilnya disimpan di hasil. Karena bil1 (2) kurang dari bil2 (3), maka hasilnya adalah 1 (true).
- **hasil = bil1 <= bil2;**: Operasi perbandingan kurang dari atau sama dengan antara bil1 dan bil2, hasilnya disimpan di hasil. Karena bil1 (2) kurang dari atau sama dengan bil2 (3), maka hasilnya adalah 1 (true).
- **hasil = bil1 == bil2;**: Operasi perbandingan sama dengan antara bil1 dan bil2, hasilnya disimpan di hasil. Karena bil1 (2) tidak sama dengan bil2 (3), maka hasilnya adalah 0 (false).
- **hasil = bil1 != bil2;**: Operasi perbandingan tidak sama dengan antara bil1 dan bil2, hasilnya disimpan di hasil. Karena bil1 (2) tidak sama dengan bil2 (3), maka hasilnya adalah 1 (true).
- **cout << hasil << endl;**: Output hasil operasi ke layar.

Output:

```

0
0
1
1
0
1

Process returned 0 (0x0)
Press any key to continue

```

4. (Operasi logika) Tuliskan kode berikut dan jalankan. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban.

Code:

```

#include <iostream>

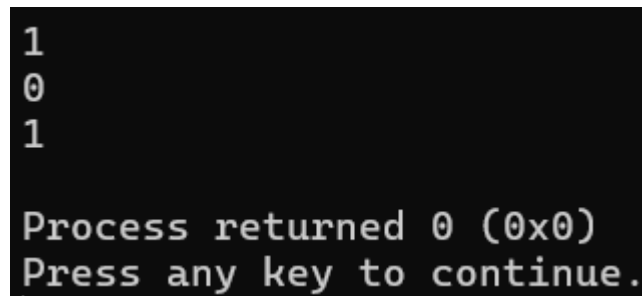
using namespace std;

int main()
{
    int bil1 = 2, bil2 = 3, hasil;
    hasil = bil1 <= bil2 and bil1 < bil2;
    cout << hasil << endl;
    hasil = bil1 >= bil2 and bil1 < bil2;
    cout << hasil << endl;
    hasil = not(bil1 >= bil2) or bil1 < bil2;
    cout << hasil << endl;
    return 0;
}

```

- **int bil1 = 2, bil2 = 3, hasil;**: Deklarasi variabel bil1 dan bil2 dengan nilai awal 2 dan 3, serta variabel hasil untuk menyimpan hasil operasi.
- **hasil = bil1 <= bil2 and bil1 < bil2;**: Operasi logika AND antara dua kondisi: **bil1 <= bil2** dan **bil1 < bil2**. Karena **bil1** (2) kurang dari atau sama dengan **bil2** (3) dan **bil1** (2) kurang dari **bil2** (3), maka hasilnya adalah 1 (true).
- **hasil = bil1 >= bil2 and bil1 < bil2;**: Operasi logika AND antara dua kondisi: **bil1 >= bil2** dan **bil1 < bil2**. Karena **bil1** (2) tidak lebih besar atau sama dengan **bil2** (3) dan **bil1** (2) kurang dari **bil2** (3), maka hasilnya adalah 0 (false).
- **hasil = not(bil1 >= bil2) or bil1 < bil2;**: Operasi logika OR antara dua kondisi: **not(bil1 >= bil2)** dan **bil1 < bil2**. Karena **bil1** (2) tidak lebih besar atau sama dengan **bil2** (3) dan **bil1** (2) kurang dari **bil2** (3), maka hasilnya adalah 1 (true).
- **cout << hasil << endl;**: Output hasil operasi ke layar.

Output:



```

1
0
1

Process returned 0 (0x0)
Press any key to continue.

```

5. (Percabangan if-else) Tuliskan kode berikut dan jalankan. Masukkan input 80, 81, dan 79. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban.

Code:

```

#include <iostream>

using namespace std;

int main()
{
    int nilai;
    cin >> nilai;
    if (nilai > 80) {
        cout << "A" << endl;
    } else {
        cout << "Bukan A" << endl;
    }
    return 0;
}

```

```

50
Bukan A

Process returned 0 (0x0)
Press any key to continue

90
A

Process returned 0 (0x0)
Press any key to continue.

```

Output

Penjelasan:

- **cin >> nilai;**: Input nilai dari pengguna menggunakan fungsi **cin**.
 - **if (nilai > 80)**: Operasi kondisional yang mengecek apakah nilai lebih besar dari 80. Jika nilai lebih besar dari 80, maka program akan menampilkan "A" ke layar. Jika nilai tidak lebih besar dari 80, maka program akan menampilkan "Bukan A" ke layar.
 - **cout << "A" << endl;** atau **cout << "Bukan A" << endl;**: Output hasil operasi kondisional ke layar.
6. (Perulangan for-to-do) Tuliskan kode berikut dan jalankan. Masukkan 1 dan 10. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban.

Code:

```

#include <iostream>

using namespace std;

int main()
{
    int a,b, bilangan;
    cout << "Masukan batas bawah: ";
    cin >> a;
    cout << "Masukan batas atas: ";
    cin >> b;
    for (bilangan = a; bilangan <= b; bilangan++){
        cout << "Bilangan" << bilangan << endl;
    }
    return 0;
}

```

- **int a, b, bilangan;**: Deklarasi variabel a dan b untuk menyimpan batas bawah dan batas atas yang diinput oleh pengguna, serta variabel bilangan untuk menyimpan nilai bilangan yang akan ditampilkan.
- **cout << "Masukan batas bawah: ";**: Output pesan untuk meminta pengguna memasukkan batas bawah. **cin >> a;**: Input batas bawah dari pengguna menggunakan fungsi **cin**. **cout << "Masukan batas atas: ";**: Output pesan untuk meminta pengguna memasukkan batas atas. **cin >> b;**: Input batas atas dari pengguna menggunakan fungsi **cin**.
- **for (bilangan = a; bilangan <= b; bilangan++)**: Operasi perulangan menggunakan loop **for** untuk menampilkan bilangan dari batas bawah hingga batas atas. **bilangan = a**: Inisialisasi nilai bilangan dengan batas bawah. **bilangan <= b**: Kondisi perulangan yang mengecek apakah nilai bilangan kurang dari atau sama dengan batas atas. **bilangan++**: Operasi increment nilai bilangan setiap kali perulangan.
- **cout << "Bilangan" << bilangan << endl;**: Output hasil perulangan ke layar, yaitu bilangan yang sedang diproses.

Output:

```

Masukan batas bawah: 15
Masukan batas atas: 10

Process returned 0 (0x0)
Press any key to continue

```

7. (Perulangan while-do) Tuliskan kode berikut dan jalankan. Masukkan pada input bilangan 10. Screenshot kode dan hasilnya, lalu tempelkan pada jawaban.

Code:


```

#include <iostream>

using namespace std;

int main()
{
    int bilangan, asli, jumlah;

    cout << "Masukkan bilangan asli: ";
    cin >> asli;

    bilangan = 1;
    jumlah = 0;
    while (bilangan <= asli){
        if (bilangan % 2 == 0){
            jumlah += bilangan;
        }
        bilangan++;
    }
    return 0;
}

```

- **int bilangan, asli, jumlah;**: Deklarasi variabel bilangan untuk menyimpan nilai bilangan yang sedang diproses, asli untuk menyimpan bilangan asli yang diinput oleh pengguna, dan jumlah untuk menyimpan hasil penjumlahan.
- **cout << "Masukkan bilangan asli: ";**: Output pesan untuk meminta pengguna memasukkan bilangan asli. **cin >> asli;**: Input bilangan asli dari pengguna menggunakan fungsi **cin**.
- **bilangan = 1;**: Inisialisasi nilai bilangan dengan 1. **jumlah = 0;**: Inisialisasi nilai jumlah dengan 0.
- **while (bilangan <= asli)**: Operasi perulangan menggunakan loop.
- **while** untuk menjumlahkan bilangan genap dari 1 hingga bilangan asli. **bilangan <= asli**: Kondisi perulangan yang mengecek apakah nilai bilangan kurang dari atau sama dengan bilangan asli. **if (bilangan % 2 == 0)**: Operasi kondisional yang mengecek apakah nilai bilangan genap. **jumlah += bilangan;**: Operasi penjumlahan nilai bilangan ke jumlah jika nilai bilangan genap. **bilangan++;**: Operasi increment nilai bilangan setiap kali perulangan.

Output:

```

Masukkan bilangan asli: 7
Process returned 0 (0x0)
Press any key to continue.

```

8. program yang menerima *input*-an dua buah bilangan betipe float, kemudian memberikan *output*-an hasil penjumlahan, pengurangan, perkalian, dan pembagian dari dua bilangan tersebut.

Code:

```

#include <iostream>
#include <limits>

int main() {
    float num1, num2;

    // Mendapatkan dua bilangan float dari pengguna
    std::cout << " Masukkan bilangan pertama: ";
    std::cin >> num1;
    std::cout << " Masukkan bilangan kedua: ";
    std::cin >> num2;

    // Menghitung hasilnya
    float penjumlahan = num1 + num2;
    float pengurangan = num1 - num2;
    float perkalian = num1 * num2;
    float pembagian = (num2 != 0) ? num1 / num2 : std::numeric_limits<float>::quiet_NaN();

    // Print the results
    std::cout << " Hasil: " << std::endl;
    std::cout << " penjumlahan: " << penjumlahan << std::endl;
    std::cout << " pengurangan: " << pengurangan << std::endl;
    std::cout << " perkalian: " << perkalian << std::endl;
    if (num2 != 0) {
        std::cout << " pembagian: " << pembagian << std::endl;
    } else {
        std::cout << " pembagian: Error: pembagian dengan 0!" << std::endl;
    }
    return 0;
}

```

- **float num1, num2;**: Deklarasi variabel num1 dan num2 untuk menyimpan dua bilangan float yang diinput oleh pengguna.
- **std::cout << " Masukkan bilangan pertama: ";**: Output pesan untuk meminta pengguna memasukkan bilangan pertama. **std::cin >> num1;**: Input bilangan pertama dari pengguna menggunakan fungsi **std::cin**.
- **std::cout << " Masukkan bilangan kedua: ";**: Output pesan untuk meminta pengguna memasukkan bilangan kedua. **std::cin >> num2;**: Input bilangan kedua dari pengguna menggunakan fungsi **std::cin**.
- **float penjumlahan = num1 + num2;**: Operasi penjumlahan antara num1 dan num2. **float pengurangan = num1 - num2;**: Operasi pengurangan antara num1 dan num2. **float perkalian = num1 * num2;**: Operasi perkalian antara num1 dan num2. **float pembagian = (num2 != 0) ? num1 / num2 :**
std::numeric_limits<float>::quiet_NaN();: Operasi pembagian antara num1 dan num2. Jika num2 tidak sama dengan 0, maka hasilnya adalah num1 dibagi num2. Jika num2 sama dengan 0, maka hasilnya adalah NaN (Not a Number) yang dihasilkan oleh fungsi **std::numeric_limits<float>::quiet_NaN()**.
- **std::cout << " Hasil: " << std::endl;**: Output pesan untuk menampilkan hasilnya. **std::cout << " penjumlahan: " << penjumlahan << std::endl;**: Output hasil penjumlahan. **std::cout << " pengurangan: " << pengurangan << std::endl;**: Output hasil pengurangan. **std::cout << " perkalian: " << perkalian << std::endl;**: Output hasil perkalian. **if (num2 != 0) { ... }**: Operasi kondisional yang mengecek apakah num2 tidak sama dengan 0. Jika num2 tidak sama dengan 0, maka output hasil pembagian. Jika num2 sama dengan 0, maka output pesan error pembagian dengan 0.

Output:

```
Masukkan bilangan pertama: 5
Masukkan bilangan kedua: 2
Hasil:
penjumlahan: 7
pengurangan: 3
perkalian: 10
pembagian: 2.5

Process returned 0 (0x0)   execut
Press any key to continue.
```

9. Buatlah sebuah program yang menerima masukan angka dan mengeluarkan *output* nilai angka tersebut dalam bentuk tulisan. Angka yang akan di- *input*-kan user adalah bilangan bulat positif mulai dari 0 s.d 100.

Code:

```
#include <iostream>
#include <string>

std::string angkaToTulisan(int angka) {
    std::string tulisan;

    if (angka < 10) {
        switch (angka) {
            case 0: tulisan = "nol"; break;
            case 1: tulisan = "satu"; break;
            case 2: tulisan = "dua"; break;
            case 3: tulisan = "tiga"; break;
            case 4: tulisan = "empat"; break;
            case 5: tulisan = "lima"; break;
            case 6: tulisan = "enam"; break;
            case 7: tulisan = "tujuh"; break;
            case 8: tulisan = "delapan"; break;
            case 9: tulisan = "sembilan"; break;
        }
    } else if (angka < 20) {
        if (angka == 10) tulisan = "sepuluh";
        else if (angka == 11) tulisan = "sebelas";
        else {
            std::string belasan = "";
            switch (angka % 10) {
                case 1: belasan = "se"; break;
                case 2: belasan = "dua"; break;
                case 3: belasan = "tiga"; break;
                case 4: belasan = "empat"; break;
                case 5: belasan = "lima"; break;
                case 6: belasan = "enam"; break;
                case 7: belasan = "tujuh"; break;
                case 8: belasan = "delapan"; break;
            }
        }
    }
}
```

```

        case 9: tulisan = "sembilan"; break;
    }
} else if (angka < 20) {
    if (angka == 10) tulisan = "sepuluh";
    else if (angka == 11) tulisan = "sebelas";
    else {
        std::string belasan = "";
        switch (angka % 10) {
            case 1: belasan = "se"; break;
            case 2: belasan = "dua"; break;
            case 3: belasan = "tiga"; break;
            case 4: belasan = "empat"; break;
            case 5: belasan = "lima"; break;
            case 6: belasan = "enam"; break;
            case 7: belasan = "tujuh"; break;
            case 8: belasan = "delapan"; break;
            case 9: belasan = "sembilan"; break;
        }
        tulisan = "belas " + belasan;
    }
} else if (angka < 100) {
    std::string puluhan = "";
    switch (angka / 10) {
        case 2: puluhan = "dua"; break;
        case 3: puluhan = "tiga"; break;
        case 4: puluhan = "empat"; break;
        case 5: puluhan = "lima"; break;
        case 6: puluhan = "enam"; break;
        case 7: puluhan = "tujuh"; break;
        case 8: puluhan = "delapan"; break;
        case 9: puluhan = "sembilan"; break;
    }
}
if (angka % 10 != 0) {
    std::string satuan = "";
    switch (angka % 10) {
        case 1: satuan = "satu"; break;
        case 2: satuan = "dua"; break;
        case 3: satuan = "tiga"; break;
        case 4: satuan = "empat"; break;
        case 5: satuan = "lima"; break;
        case 6: satuan = "enam"; break;
        case 7: satuan = "tujuh"; break;
        case 8: satuan = "delapan"; break;
        case 9: satuan = "sembilan"; break;
    }
    tulisan = puluhan + " puluh " + satuan;
} else {
    tulisan = puluhan + " puluh";
}
}
return tulisan;
}

int main() {
    int angka;
    std::cout << "Masukkan angka (0-100): ";
    std::cin >> angka;

    if (angka >= 0 && angka <= 100) {
        std::cout << "Angka " << angka << " dalam bentuk tulisan adalah: " << angkaToTulisan(angka) << std::endl;
    } else {
        std::cout << "Angka tidak valid!" << std::endl;
    }

    return 0;
}

```

- Fungsi **angkaToTulisan** adalah fungsi yang mengubah angka menjadi tulisan. Fungsi ini memiliki satu parameter, yaitu **angka**, yang merupakan angka yang ingin diubah menjadi tulisan. Jika **angka** kurang dari 10, maka fungsi menggunakan **switch** untuk mengubah angka menjadi tulisan. Jika **angka** antara 10 dan 20, maka fungsi menggunakan **if** dan **switch** untuk mengubah angka menjadi tulisan. Jika **angka** antara 20 dan 100, maka fungsi menggunakan **switch** untuk mengubah puluhan dan satuan menjadi tulisan.
- Fungsi **main** adalah fungsi utama yang dijalankan oleh program. Fungsi ini memiliki beberapa langkah, yaitu: Membaca angka dari pengguna menggunakan **std::cin**. Memeriksa apakah angka valid (antara 0 dan 100). Jika angka valid, maka fungsi **angkaToTulisan** dipanggil untuk mengubah angka menjadi tulisan. Hasil pengubahan angka menjadi tulisan ditampilkan ke layar menggunakan **std::cout**.

Output:

```
Masukkan angka (0-100): 56
Angka 56 dalam bentuk tulisan adalah: lima puluh enam

Process returned 0 (0x0)   execution time : 2.785 s
Press any key to continue.
```

10. Buatlah program yang dapat memberikan input dan output sbb.

```
input: 3
output:
  3 2 1 * 1 2 3
    2 1 * 1 2
      1 * 1
        *
```

Code:

```
#include <iostream>

void print_pattern(int n) {
    for (int i = 1; i <= n; i++) {
        // Print leading spaces
        for (int j = 1; j <= i; j++) {
            std::cout << " ";
        }

        // Print numbers from n down to 1
        for (int j = n - i + 1; j >= 1; j--) {
            std::cout << j << " ";
        }

        // Print asterisk
        std::cout << "* ";

        // Print numbers from 1 up to n
        for (int j = 1; j <= n - i + 1; j++) {
            std::cout << j << " ";
        }

        std::cout << std::endl;
    }
}

int main() {
    int n;
    std::cout << "Masukkan angka: ";
    std::cin >> n;
    print_pattern(n);
    return 0;
}
```

- Fungsi **print_pattern** adalah fungsi yang mencetak pola angka dan asterisk (*). Fungsi ini memiliki satu parameter, yaitu **n**, yang merupakan angka yang menentukan ukuran pola.
- Pola yang dicetak oleh fungsi **print_pattern** terdiri dari beberapa bagian: mencetak spasi-spasi yang menentukan indentasi pola. mencetak angka-angka dari **n** ke 1. mencetak asterisk (*). mencetak angka-angka dari 1 ke **n**.

Output:

Masukkan angka: 3

3 2 1 * 1 2 3

2 1 * 1 2

1 * 1

Process returned 0 (0x0)

Press any key to continue.