

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL 1  
CODE BLOCKS IDE &  
PENGENALAN BAHASA C++ (BAGIAN PERTAMA)**



**Disusun Oleh :**

NAMA : Devi Nurliana

NIM : 103112400144

**Dosen**

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

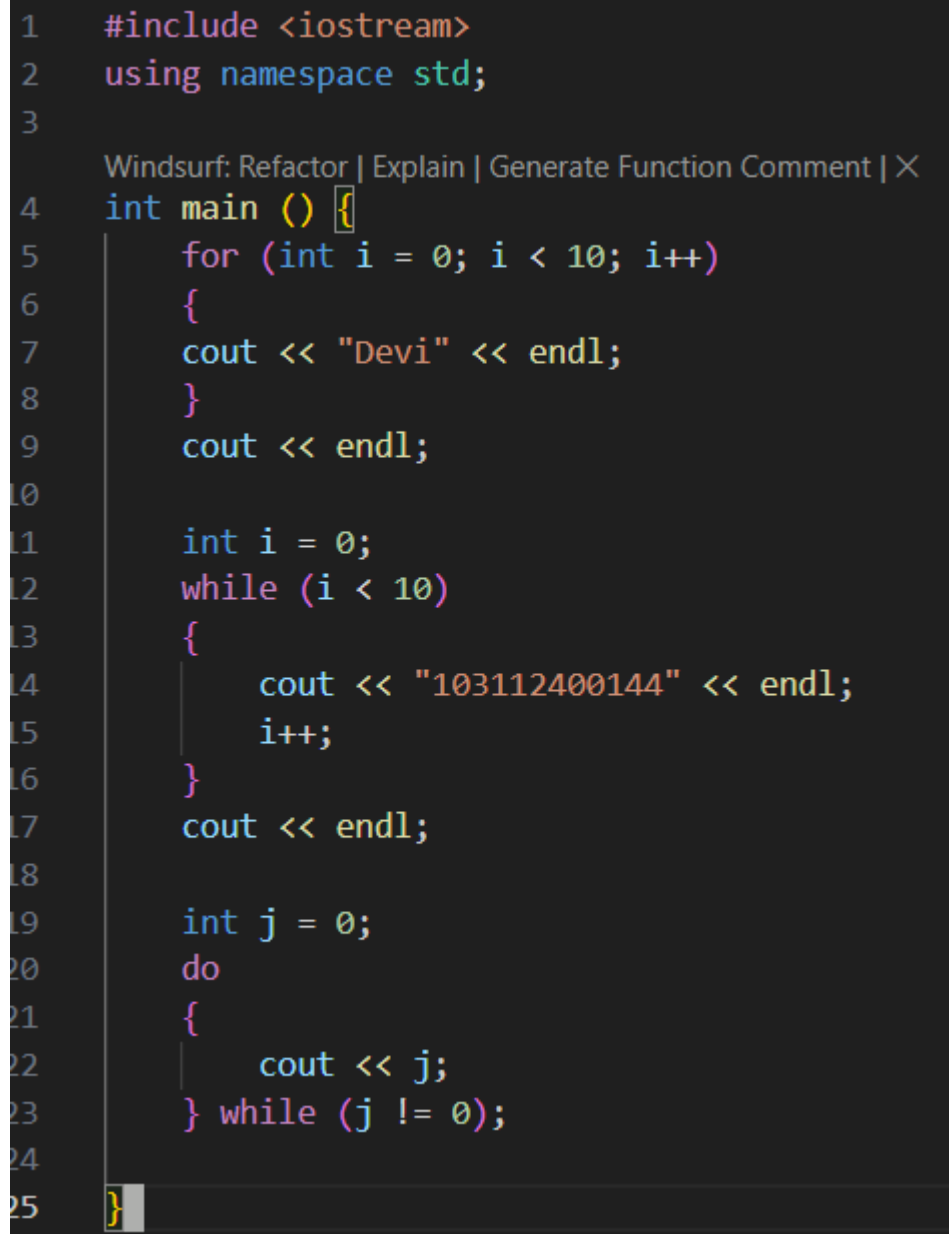
## A. Dasar Teori

C++ adalah pengembangan dari bahasa c yang dibuat oleh Bjarne Stroustrup sekitar tahun 1980-an. C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (pakai fungsi biasa), berorientasi objek (pakai class dan object), atau bahkan gabungan keduanya. C++ punya dasar-dasar seperti variabel, operator percabangan (if, switch), perulangan (for, while), dan bisa memakai class untuk membuat objek.

## B. Guided (berisi screenshot source code & output program disertai penjelasannya)

### 1. Guide 1

#### a. Source Code



```
1  #include <iostream>
2  using namespace std;
3
4  int main () {
5      for (int i = 0; i < 10; i++)
6      {
7          cout << "Devi" << endl;
8      }
9      cout << endl;
10
11     int i = 0;
12     while (i < 10)
13     {
14         cout << "103112400144" << endl;
15         i++;
16     }
17     cout << endl;
18
19     int j = 0;
20     do
21     {
22         cout << j;
23     } while (j != 0);
24
25 }
```

#### b. Screenshot Output

```
Devi
Devi
Devi
Devi
Devi
Devi
Devi
Devi
Devi
Devi
Devi

103112400144
103112400144
103112400144
103112400144
103112400144
103112400144
103112400144
103112400144
103112400144
103112400144
```

c. Deskripsi

Program ini berfungsi untuk menampilkan teks dan angka melalui tiga tipe perulangan, dengan langkah kerja sebagai berikut:

1. Perulangan dengan for  
Program menggunakan `for (int i = 0; i < 10; i++)` untuk mencetak kata “Devi” sebanyak 10 kali, tiap baris satu kata. Setelah selesai, dicetak baris kosong.
2. Perulangan dengan while  
Program menetapkan `i = 0` lalu `while (i < 10)` mencetak “103112400144” sebanyak 10 kali, tiap baris satu kali. Setelah selesai, dicetak baris kosong.
3. Perulangan dengan do-while  
Program menetapkan `j = 0` lalu menjalankan `do { cout << j; } while (j != 0);`. Karena kondisi `j != 0` bernilai false sejak awal (karena `j = 0`), maka blok `do` tetap dijalankan sekali → mencetak 0. Setelah itu, perulangan berhenti.
4. Selesai  
Program kemudian berakhir dan mengembalikan nilai 0.

2. Guide 2

a. Source Code

```

1  #include <iostream>
2  using namespace std;
3
4  Windsurf: Refactor | Explain | Generate Function Comment | X
5  int main () {
6      int hari;
7      cout << "hari 1-7" ;
8      cin >> hari;
9      if (hari == 7)
10     {
11         cout << "hari senin\n";
12     }
13     else
14     {
15         cout << "hari kerja\n";
16     }
17
18     switch (hari)
19     {
20         case 7:
21             cout << "hari selasa\n";
22             break;
23
24             default:
25                 cout << "hari kerja\n";
26                 break;
27     }
28     return 0;
29 }
30
31

```

b. Screenshot Output

```

Microsoft-MIEngine-Error: Microsoft-MIEngine-Error:
vme0ph.4ir' '--pid=Microsoft-MIEngine-Pid-tco4wptd.wfq' '--dbgExe=C:\T
-GCC-64\bin\gdb.exe' '--interpreter=mi'
hari 1-7

```

c. Deskripsi

Program ini berfungsi untuk mengenali hari menurut input angka, dengan langkah kerja sebagai berikut:

1. Input Hari  
Program mencetak “hari 1-7” lalu meminta pengguna memasukkan nilai `hari` (integer).
2. Struktur `if`
  - Jika `hari == 7`, program mencetak “hari senin”.
  - Jika tidak, program mencetak “hari kerja”.
3. Struktur `switch`
  - Untuk `case 7`, program mencetak “hari senin”.
  - Untuk `default` (nilai selain 7), program mencetak “hari kerja”.
4. Selesai  
Program mengembalikan nilai 0 dan berhenti.

### 3. Guide 3

#### a. Source Code

```

1  #include <iostream>
2  using namespace std;
3
4  struct mahasiswa
5  {
6      string nama;
7      string nim;
8  };
9
10 int main()
11 {
12     mahasiswa mhs;
13     mhs.nama = "Devi";
14     mhs.nim = "103112400144";
15
16     cout << "nama :" << mhs.nama << endl;
17     cout << "nim :" << mhs.nim << endl;
18
19 }
```

#### b. Screenshot Output

```

2wzio2.e0m' '--pid=Microsoft-MIEngine-Pid-aten
-GCC-64\bin\gdb.exe' '--interpreter=mi'
nama :Devi
nim :10311240014
```

#### c. Deskripsi

Program ini berfungsi untuk menyimpan dan menampilkan data mahasiswa sederhana melalui struct, dengan langkah kerja sebagai berikut:

1. Definisi struct mahasiswa  
Struktur mahasiswa dibuat dengan dua anggota: nama (tipe string) dan nim (tipe string).
2. Inisialisasi data  
Di dalam main(), dibuat variabel mhs bertipe mahasiswa. Nilai nama diisi dengan “Devi” dan nim diisi dengan “10311240014”.
3. Menampilkan data  
Program mencetak “nama :Devi” dan “nim :10311240014” ke layar, masing-masing di baris baru.
4. Selesai  
Program berakhir setelah menampilkan data.

#### 4. Guide 4

##### a. Source Code

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  Windsurf: Refactor | Explain
6  struct mahasiswa
7  {
8      string nama;
9      string nim;
10 };
11
12 Windsurf: Refactor | Explain | Generate Function Comment | X
13 int main()
14 {
15     mahasiswa mhs;
16     // mhs.nama = "Devi"
17     // mhs.nim = "103112400144";
18
19     cout << "nama :";
20     getline(cin, mhs.nama);
21     cout << "nim :";
22     cin >> mhs.nim;
23
24     cout << "nama :" << mhs.nama << endl;
25     cout << "nim :" << mhs.nim << endl;
26 }

```

b. Screenshot Output

```

nama :Devi
nim :103112400144
nama :Devi
nim :103112400144

```

c. Deskripsi

Program ini berfungsi untuk menyimpan dan menampilkan data mahasiswa dengan input dari pengguna, melalui struct, dengan langkah kerja sebagai berikut:

1. Definisi struct mahasiswa  
Struktur mahasiswa dibuat dengan dua anggota: nama (tipe string) dan nim (tipe string).
2. Input data
  - Program mencetak “nama :” lalu membaca input nama menggunakan `getline(cin, mhs.nama)` agar nama yang mengandung spasi bisa terbaca secara utuh.
  - Kemudian mencetak “nim :” lalu membaca input nim menggunakan `cin >> mhs.nim`.
3. Menampilkan data  
Program mencetak “nama :” diikuti isi `mhs.nama` dan baris baru, lalu mencetak “nim :” diikuti isi `mhs.nim`.
4. Selesai  
Program kemudian berhenti (akhir dari main).

## 5. Guide 5

### a. Source Code

```
1  #include <iostream>
2  using namespace std;
3
4  Windsurf: Refactor | Explain | Generate Function Comment | X
5  int main() {
6      float x, y;
7      cin >> x >> y;
8      cout << "Penjumlahan: " << x + y << endl;
9      cout << "Pengurangan: " << x - y << endl;
10     cout << "Perkalian: " << x * y << endl;
11     if (y != 0) {
12         cout << "Pembagian: " << x / y << endl;
13     } else {
14         cout << "Pembagian: tak terdefinisi" << endl;
15     }
16     return 0;
17 }
```

### b. Screenshot Output

```
6
2
Penjumlahan: 8
Pengurangan: 4
Perkalian: 12
Pembagian: 3
```



### c. Deskripsi

Program ini berfungsi sebagai kalkulator mini untuk dua bilangan desimal dengan langkah kerja sebagai berikut:

1. Input Bilangan  
Program membaca dua bilangan desimal dari pengguna ke dalam variabel x dan y menggunakan cin.
2. Operasi Aritmatika  
Program menampilkan hasil dari:
  - o Penjumlahan ( $x + y$ )
  - o Pengurangan ( $x - y$ )
  - o Perkalian ( $x * y$ )
3. Pembagian dengan Pengecekan Nol  
Program memeriksa apakah  $y \neq 0$ .
  - o Jika benar, maka program menampilkan hasil pembagian ( $x / y$ ).
  - o Jika  $y == 0$ , program menampilkan pesan "Pembagian: tak terdefinisi".
4. Selesai  
Program mengembalikan nilai 0 dan berakhir.

## 6. Guide 6

### a. Source Code

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  Windsurf: Refactor | Explain | Generate Function Comment | X
6  string angkaKeTeks(int n) {
7      string satuan[] = { "nol", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan",
8                          "sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas",
9                          "enam belas", "tujuh belas", "delapan belas", "sembilan belas" };
10     string puluhan[] = { "", "sepuluh", "dua puluh", "tiga puluh", "empat puluh", "lima puluh",
11                          "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh" };
12     if (n < 20) {
13         return satuan[n];
14     } else if (n < 100) {
15         int p = n / 10;
16         int s = n % 10;
17         if (s == 0) {
18             return puluhan[p];
19         } else {
20             return puluhan[p] + " " + satuan[s];
21         }
22     } else if (n == 100) {
23         return "seratus";
24     } else {
25         return "di luar jangkauan";
26     }
27 }
28
29 Windsurf: Refactor | Explain | Generate Function Comment | X
30 int main() {
31     int x;
32     cin >> x;
33     cout << angkaKeTeks(x) << endl;
34     return 0;
35 }
```

### b. Screenshot Output

```
12
dua belas
```

### c. Deskripsi

Program ini berfungsi untuk mengubah angka bulat 0 sampai 100 menjadi kata, dengan langkah kerja sebagai berikut:

1. Input Angka

Program membaca satu bilangan bulat  $x$  dari pengguna menggunakan `cin`.

2. Fungsi angkaKeTeks

- Bila  $x < 20$ , fungsi langsung mengembalikan kata dari array `satuan[x]`.
- Bila  $20 \leq x < 100$ , fungsi menghitung puluhan  $p = x / 10$  dan satuan  $s = x \% 10$ .
  - Jika  $s == 0$ , mengembalikan kata puluhan saja (misalnya “dua puluh”).
  - Jika  $s != 0$ , mengembalikan gabungan “puluhan + satuan” (misalnya “dua puluh satu”).
- Bila  $x == 100$ , mengembalikan “seratus”.
- Bila  $x$  di luar rentang 0–100, mengembalikan “di luar jangkauan”.

3. Menampilkan Hasil

Setelah memperoleh hasil teks dari `angkaKeTeks(x)`, program mencetak hasil tersebut ke layar.

4. Selesai

Program kemudian berhenti dengan akhir fungsi `main`.

7. Guide 7

a. Source Code

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {      int n;
5      cout << "Masukkan angka: ";
6      cin >> n;
7
8      for (int i = n; i >= 1; i--) {
9
10         for (int spasi = 0; spasi < n - i; spasi++) {
11             cout << " ";
12         }
13
14         for (int j = i; j >= 1; j--) {
15             cout << j << " ";
16         }
17
18         cout << "* ";
19
20         for (int j = 1; j <= i; j++) {
21             cout << j << " ";
22         }
23
24         cout << endl;
25     }
26
27     for (int spasi = 0; spasi < n; spasi++) {
28         cout << " ";
29     }
30     cout << "*" << endl;
31
32     return 0;
33 }

```

b. Screenshot Output

```

Masukkan angka: 3
3 2 1 * 1 2 3
  2 1 * 1 2
    1 * 1
      *

```

c. Deskripsi

Program ini berfungsi untuk mencetak pola segitiga sama sisi dari besar ke kecil dengan karakter angka dan simbol \*, dengan langkah kerja sebagai berikut:

1. Input Angka  
Program mencetak “Masukkan angka: ” lalu membaca nilai  $n$  dari pengguna.
2. Perulangan Baris Utama  
Untuk  $i$  dari  $n$  turun ke 1:
  - Cetak spasi sebanyak  $n - i$  (dua spasi per satu spasi “kosong”) untuk menggeser pola ke kanan.
  - Cetak angka menurun dari  $i$  ke 1, setiap angka diikuti satu spasi.
  - Cetak \* (simbol bintang dan satu spasi).
  - Cetak angka menaik dari 1 ke  $i$ , setiap angka diikuti satu spasi.
  - Baris baru.
3. Baris Terakhir (puncak segitiga)  
Setelah pola utama, cetak spasi sebanyak  $n$  kali (dua spasi per spasi) untuk posisi tengah, lalu cetak \*.
4. Selesai  
Program kemudian berhenti (mengembalikan nilai 0).

### C. Kesimpulan

Dalam praktikum Modul 1 ini, saya memahami secara langsung bagaimana C++ bekerja dari dasar: membuat perulangan (for, while, do-while) untuk menjalankan instruksi berulang, dan menggunakan percabangan (if-else, switch) untuk menentukan jalur eksekusi berdasarkan kondisi tertentu. Saya juga menerapkan penggunaan `struct` untuk menyimpan data seperti nama dan NIM baik melalui inisialisasi langsung maupun input pengguna.

Selain itu, saya belajar melakukan operasi aritmatika dasar (penjumlahan, pengurangan, perkalian, pembagian), lengkap dengan pengecekan agar tidak terjadi pembagian oleh nol. Tidak kalah penting, saya mempraktikkan bagaimana mengubah angka menjadi kata menggunakan array dan percabangan, serta mencetak pola angka berbentuk segitiga simetris dengan perulangan bersarang.

Walaupun konsep percabangan dan perulangan bersarang sedikit membingungkan pada awalnya, latihan langsung selama praktikum memudahkan saya memahami alur logika dan struktur kode. Keseluruhan modul ini menjadi fondasi yang kuat untuk membangun kemampuan dalam bersusun program yang lebih kompleks di kemudian hari.

### D. Referensi

- Jenis-jenis Operator Aritmatika Bahasa C++” — Duniaikom
- C++ Programs To Create Pyramid and Pattern” — Programiz