

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : ZAKI MAULA DHIA

NIM : 103112400127

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)

Guided 1

```
#include <iostream>
using namespace std;
int main()
{
    int hari;
    cout << "hari 1-7: ";
    cin >> hari;
    if (hari == 7)
    {
        cout << "hari minggu\n";
    }
    else{
        cout << "hari kerja\n";
    }
    switch (hari)
    {
        case 7:
            cout << "hari minggu\n";
            break;
        default:
            cout << "hari kerja\n";
            break;
    }
}
```

Screenshots Output

```

PS C:\Users\MyBook Hype AMD\Documents\strukturdata> & 'c:\windows\WinSxS\x-ww7-qntn\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-103112400127-or-0akjfbro.rsc' '--pid=Microsoft-MIEngine-Pid-2ru13ygy.2
hari 1-7: 2
hari kerja
hari kerja
PS C:\Users\MyBook Hype AMD\Documents\strukturdata> █

```

Deskripsi:

Program ini berfungsi sebagai pengingat hari kerja, dengan langkah kerja sebagai berikut:

1. **Input Bilangan:** Program meminta pengguna untuk memasukkan **satu bilangan integer**. Bilangan tersebut akan disimpan di variable "hari".
2. **FUNGSI IF-ELSE:** Program membandingkan variable dengan kondisi yang diminta dan menampilkan hasilnya.
3. **FUNGSI SWITCH-CASE:** Fungsi SWITCH-CASE akan memilih kondisi mana yang sesuai dengan variable "hari".
4. **Selesai:** Program akan langsung berhenti.

Guided 2

```

#include <iostream>
using namespace std;
int main()
{
    for (int i =0; i < 10; i++)
    {
        cout << "zaki maula dhia\n";
    }
    cout << endl;

    int i = 0;
    while (i < 10)
    {
        cout << "103112400127\n";
        i++;
    }
    cout << endl;

    int j = 0;
    do
    {
        cout << j;
    } while (j != 0);
}

```

```
}
```

Screenshots Output

```
PS C:\Users\MyBook Hype AMD\Documents\strukturdata> &
n\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine
or-wnwbexz.ss2' '--pid=Microsoft-MIEngine-Pid-vzdkgwq
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia
zaki maula dhia

103112400127
103112400127
103112400127
103112400127
103112400127
103112400127
103112400127
103112400127
103112400127
103112400127
103112400127

0
```

Deskripsi:

Program ini akan melakukan perulangan sesuai value yang diminta, dengan langkah berikut:

1. **FUNGSI FOR:** Fungsi ini akan mengulang perintah yang diberikan sampai kondisinya tercapai.
2. **FUNGSI WHILE-DO:** Fungsi ini akan mengulang perintah yang diberikan dengan syarat suatu kondisi harus terpenuhi terlebih dahulu.
3. **FUNGSI DO-WHILE:** Fungsi ini akan mengulang perintah yang diberikan sampai kondisinya tidak benar.

Guided 3

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

struct Mahasiswa
{
    string nama;
    string NIM;
};

int main()
{
    Mahasiswa mhs;
    mhs.nama = "zaki maula dhia";
    mhs.NIM = "103112400127";

    cout << "Nama: " << mhs.nama << endl
         << "NIM : " << mhs.NIM;
}
```

Screenshots Output

```
PS C:\Users\MyBook Hype AMD\Documents\strukturdata> & 'n\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-or-1zu4rb1e.xu3' '--pid=Microsoft-MIEngine-Pid-ilgd1240.
Nama: zaki maula dhia
NIM : 103112400127
PS C:\Users\MyBook Hype AMD\Documents\strukturdata>
```

Deskripsi:

Program ini akan mempermudah dev, jika suatu program memiliki banyak variable yang sama untuk setiap classnya.

1. **STRUCT** : Menyimpan nama variable agar bisa digunakan lagi di class lain.
2. **"Mahasiswa mhs;"** : berfungsi untuk naming struct tadi agar program dapat membacanya.
3. **OUTPUT** : Dengan memanggil inisial struct dan salah satu variable didalam struct

Guided 4

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

struct Mahasiswa
{
    string nama;
    string NIM;
};

int main()
{
    Mahasiswa mhs;
    //mhs.nama = "zaki maula dhia";
    //mhs.NIM = "103112400127";
    cout << "Nama : ";
    getline(cin, mhs.nama);
    cout << "NIM : ";
    cin >> mhs.NIM;

    cout << "Nama: " << mhs.nama << endl
         << "NIM : " << mhs.NIM;
}
```

Screenshots Output

```
PS C:\Users\MyBook Hype AMD\Documents\strukturdata> &
n\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine
or-q0xrqlac.5il' '--pid=Microsoft-MIEngine-Pid-0pm5cznd
Nama : zaki maula dhia
NIM : 103112400127
Nama: zaki maula dhia
NIM : 103112400127
PS C:\Users\MyBook Hype AMD\Documents\strukturdata> █
```

Deskripsi:

Program ini akan mempermudah dev, jika suatu program memiliki banyak variable yang sama untuk setiap classnya.

1. **STRUCT** : Menyimpan nama variable agar bisa digunakan lagi di class lain.
2. **“Mahasiswa mhs;”** : berfungsi untuk naming struct tadi agar program dapat membacanya.
3. **INPUT USER** : Menyimpan data di variable dalam struct
4. **OUTPUT** : Dengan memanggil inisial struct dan salah satu variable didalam

struct

Guided 5

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Nama : zaki maula dhia\nNIM : 103112400127" << endl;

    int nomor =1;
    cout << nomor << endl;
    cout << nomor++ << endl;
    cout << ++nomor << endl;
    return 0;
}
```

Screenshots Output

```
PS C:\Users\MyBook Hype AMD\Documents\strukturdata> & 'n\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-gzrfjpyj.iej' '--pid=Microsoft-MIEngine-Pid-xfubjxx.
Nama : zaki maula dhia
NIM : 103112400127
1
1
3
PS C:\Users\MyBook Hype AMD\Documents\strukturdata> █
```

Deskripsi:

Hanya untuk menampilkan output dan increment sederhana.

1. **OUTPUT** : Akan menampilkan semua cout termasuk yang berisikan hanya variable “nomor”.
2. **INCREMENT** : Salah dua cout memiliki perintah untuk menampilkan variable “nomor” tetapi ada “++” disebelah kanan atau kirinya, ini disebelah kanan namanya post increment sedangkan disebelah kiri pre increment, jadinya cout kedua akan menambah pada variable setelah cout menampilkan kemudian variable tersebut diteruskan ke cout ketiga dan karena cout ketiga adalah pre increment maka hasilnya akan langsung di tampilkan oleh cout.

D. Unguided (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```
#include <iostream>
using namespace std;
int main()
{
    float bilsatu;
    float bildua;

    cout << "Masukan bilangan pertama : ";
    cin >> bilsatu;
    cout << "Masukan bilangan kedua : ";
    cin >> bildua;

    cout << "Penjumlahan kedua bilangan = " << bilsatu + bildua << endl;
    cout << "Pengurangan kedua bilangan = " << bilsatu - bildua << endl;
    cout << "Perkalian kedua bilangan = " << bilsatu * bildua << endl;

    if (bildua != 0){
        cout << "Pembagian kedua bilangan = " << bilsatu / bildua <<
endl;
    }
    else{
        cout << "Pembagian dengan nol tidak terdefiniskan" << endl;
    }
}
```

Screenshots Output

```
PS C:\Users\MyBook Hype AMD\Documents\strukturdata\modul 1test> &
rs\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-af
-5orbkhhcz.icr' '--pid=Microsoft-MIEngine-Pid-lis4fsfb.sru' '--dbgE
Masukan bilangan pertama : 10
Masukan bilangan kedua : 3
Penjumlahan kedua bilangan = 13
Pengurangan kedua bilangan = 7
Perkalian kedua bilangan = 30
Pembagian kedua bilangan = 3.33333
PS C:\Users\MyBook Hype AMD\Documents\strukturdata\modul 1test> |
```

Deskripsi:

Menghitung kedua bilangan yang diinput user ke berbagai operasi matematika seperti

penjumlahan, pengurangan, perkalian, dan pembagian.

1. **INPUT** : Program akan meminta user memasukan dua bilangan genap dan meyimpan kedua bilangan tersebut di variable “bilsatu” dan “bildua”.
2. **OPERASI MATEMATIKA** : Program mulai menghitung hasil dari penjumlahan, pengurangan, perkalian, dan pembagian dari kedua bilangan tersebut secara instan.
3. **OUTPUT** : Program akan menampilkan hasil dari penjumlahan, pengurangan, perkalian, dan pembagian dari kedua bilangan tersebut ke dalam tampilan teks yang mudah dipahami oleh manusia.

Unguided 2

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

int main() {
    string satuan[] = {"nol", "satu", "dua", "tiga", "empat", "lima",
"enam", "tujuh", "delapan", "sembilan"};
    string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas",
"empat belas", "empat belas", "lima belas", "enam belas", "tuuh belas",
"delapan belas", "sembilan belas"};
    string puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat
puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh",
"sembilan puluh"};

    int angka;
    cout << "Masukan angka : ";
    cin >> angka;

    if (angka<10)
    {
        cout << satuan[angka];
    }
    else if (angka < 20){
        cout << belasan[angka-10];
    }
    else if (angka < 100){
        int p = angka / 10;
        int s = angka % 10;
        cout << puluhan[p];
        if (s != 0) cout << " " << satuan[s];
    }
    else if (angka = 100){
```

```

        cout << "seratus";
    }
    else{
        cout << "Input tidak valid";
    }
}

```

Screenshots Output

```

PS C:\Users\MyBook Hype AMD\Documents\strukturdata\modul 1test> &rs\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-44-sfilygnd.emj' '--pid=Microsoft-MIEngine-Pid-1qwq5jpq.rta' '--dbgE
Masukan angka : 43
empat puluh tiga
PS C:\Users\MyBook Hype AMD\Documents\strukturdata\modul 1test>

```

Deskripsi:

Program untuk menampilkan kata dari suatu angka dari 0-100.

1. **INPUT** : Program meminta satu bilangan dari 0-100 untuk ditampilkan bagaimana cara menulis katanya.
2. **OPERASI** : Jika program menerima bilangan kurang dari 10 akan menampilkan salah satu dari isi slice “satuan” yang dimulai dari index 0, ketika program menerima bilangan belasan kurang dari 20 akan menampilkan salah satu isi slice “belasan” lalu bilangan tersebut akan dikurangi 10 untuk mempermudah program memilih index, ketika program menerima bilangan puluhan akan menampilkan salah satu isi dari slice “puluhan” dan “satuan” secara bersamaan.
3. **OUTPUT** : Program akan menampilkan hasilnya berupa teks yang dapat dicopy.

Unguided 3

```

#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Masukan Tinggi : ";
    if (!(cin >> n) || n < 0)
    {
        cout << "Tinggi tidak valid\n";
        return 1;
    }
    for (int i = n; i >= 1; --i)

```

```

{
    int spasiDepan = n - i;
    for (int s = 0; s < spasiDepan; ++s) cout << " ";
    for (int a = i; a >= 1; --a) cout << a;
    cout << "*";
    for (int b = 1; b <= i; ++b) cout << b;
    cout << "\n";
}
for (int s = 0; s < n; ++s) cout << " ";
cout << "*" << "\n";
return 0;
}

```

Screenshots Output

```

PS C:\Users\MyBook Hype AMD\Documents\strukturdata\modul 1test> & 'rs\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ove0-gwpkcupp.3db' '--pid=Microsoft-MIEngine-Pid-japdetst.mkq' '--dbgExe
Masukan Tinggi : 4
4321*1234
 321*123
   21*12
    1*1
     *

```

Deskripsi:

Program untuk menampilkan kata dari suatu angka dari 0-100.

1. **INPUT** : Program meminta satu bilangan bulat sebagai tinggi dari segitiga dan menyimpannya di variable “tinggi”.
2. **OPERASI** : Program akan membuat perulangan untuk membuat sebuah segitiga yang dibuat dari bilangan “tinggi” dan sebuah “*”, secara urut dari yang terbesar ke yang terkecil kemudian dipisah dengan tanda “*” lalu memulai lagi tetapi dari yang terkecil ke yang terbesar.
3. **OUTPUT** : Hasil akhir dari program tersebut adalah sebuah segitiga sama kaki dengan rotasi menghadap ke bawah.