

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

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



Disusun Oleh :

NAMA : Loh Suwargi Nitis Hamengku Bintang
NIM : 103112400116

Dosen

FAHRUDIN MUKTI WIBOWO, S.Kom., M.Eng

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

A. Dasar Teori

Pada dasarnya, struktur data adalah tata susun data yang berisi kolom-kolom, baik yang terlihat oleh pengguna maupun yang hanya dipakai secara internal untuk tujuan pemrograman. Sekelompok kolom pada satu baris disebut record atau catatan. Lebar kolom dalam struktur data dapat bervariasi; beberapa kolom memiliki ukuran tetap, sedangkan lainnya dapat berubah mengikuti input yang dimasukkan pengguna(jodi;2020).

Dalam praktik pemrograman, keputusan untuk menggunakan jenis struktur data tertentu akan berdampak langsung pada efektivitas program, kerumitan penulisan kode, dan penggunaan sumber daya komputer. Struktur data tidak sebatas pada bentuk sederhana seperti array atau list, melainkan juga mencakup bentuk yang lebih maju, antara lain stack, queue, tree, dan graph. Tiap struktur data mempunyai ciri khas, kelebihan, sekaligus keterbatasan yang menentukan bagaimana data dapat diproses, dimodifikasi, dan diakses kembali.

C++ merupakan bahasa pemrograman yang diperkenalkan oleh Bjarne Stroustrup pada tahun 1983 di Bell Labs. Bahasa ini banyak dimanfaatkan dalam pengembangan perangkat lunak karena memiliki performa tinggi serta fleksibilitas dalam mengelola memori. C++ juga sangat populer untuk membangun aplikasi desktop, permainan, maupun sistem embedded. Dengan sintaks yang serupa dengan bahasa C namun diperkaya dengan berbagai fitur tambahan, C++ sangat sesuai untuk mengimplementasikan konsep-konsep struktur data, baik sederhana maupun kompleks, secara efisien.

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

Guided 1

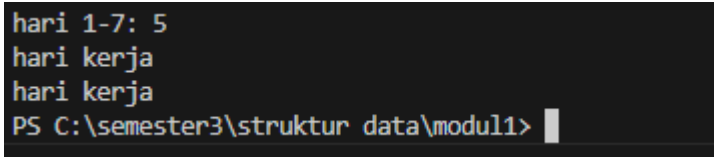
```
#include <iostream>
using namespace std;
int main()
{
    int hari;
    cout << "hari 1-7: ";
    cin >> hari;
    if (hari == 7)
    {
        cout << "hari minggu\n";
    }
    else{
        cout << "hari kerja\n";
    }
}
```

```

switch (hari)
{
case 7:
    cout << "hari minggu\n";
    break;
default:
    cout << "hari kerja\n";
    break;
}
}

```

Screenshots Output



```

hari 1-7: 5
hari kerja
hari kerja
PS C:\semester3\struktur data\modul1>

```

Deskripsi:

Program tersebut berfungsi untuk menentukan apakah input angka 1–7 merupakan hari Minggu atau hari kerja. Jika pengguna memasukkan angka 7, maka akan ditampilkan "hari Minggu", sedangkan untuk angka selain itu akan ditampilkan "hari kerja". Logika ini ditulis menggunakan dua cara, yaitu dengan struktur if–else dan switch–case.

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

Guided 2

```

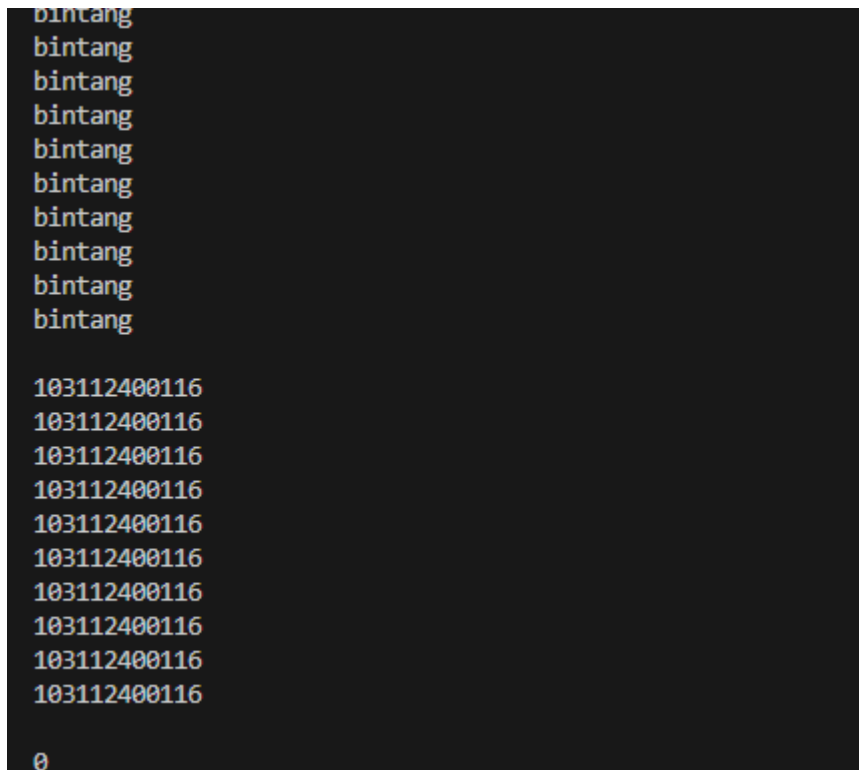
#include <iostream>
using namespace std;
int main()
{
    for (int i=0; i < 10; i++)
    {
        cout << "bintang\n";
    }
    cout << endl;

    int i = 0;
    while (i < 10)
    {
        cout << "103112400116\n";
        i++;
    }
    cout << endl;
}

```

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

Screenshots Output



```
bintang
bintang
bintang
bintang
bintang
bintang
bintang
bintang
bintang
bintang

103112400116
103112400116
103112400116
103112400116
103112400116
103112400116
103112400116
103112400116
103112400116
103112400116

0
```

Deskripsi:

Program di atas menunjukkan contoh penggunaan tiga jenis perulangan dalam C++. Pertama, perulangan for digunakan untuk menampilkan kata "bintang" sebanyak 10 kali. Kedua, perulangan while dipakai untuk mencetak angka 103112400116 sebanyak 10 kali. Terakhir, perulangan do-while menampilkan angka 0 satu kali saja karena kondisi langsung berhenti setelah dijalankan sekali. Program ini sederhana dan mudah dipahami karena memperlihatkan perbedaan cara kerja masing-masing jenis perulangan.

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

Guided 3

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

struct Mahasiswa
{
```

```

    string nama;
    string NIM;

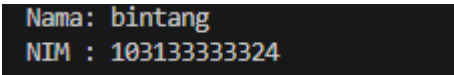
};

int main()
{
    Mahasiswa mhs;
    mhs.nama = "bintang";
    mhs.NIM = "10313333324";

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

```

Screenshots Output



```

Nama: bintang
NIM : 10313333324

```

Deskripsi:

Program di atas menggunakan struct untuk merepresentasikan data mahasiswa. Struct bernama Mahasiswa memiliki dua atribut, yaitu nama dan NIM. Pada fungsi main, dibuat sebuah objek mhs dengan nilai nama “bintang” dan NIM “10313333324”. Kemudian, program menampilkan data tersebut ke layar dalam format: Nama: bintang dan NIM: 10313333324. Program ini sederhana namun berguna sebagai contoh penyimpanan dan pengaksesan data menggunakan struct.

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

Guided 4

```

#include <iostream>
using namespace std;

struct Mahasiswa
{
    string nama;
    string NIM;
};

int main()
{
    Mahasiswa mhs;
    //mhs.nama = "bintang";
    //mhs.NIM = "10313333324";
}

```

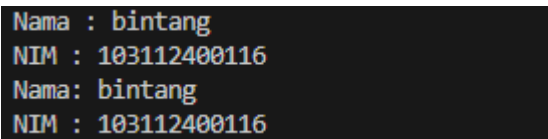
```

cout << "Nama : ";
getline(cin, mhs.nama);
cout << "NIM : ";
cin >> mhs.NIM;

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

```

Screenshots Output



```

Nama : bintang
NIM : 103112400116
Nama: bintang
NIM : 103112400116

```

Deskripsi:

Program di atas menggunakan struct Mahasiswa yang berisi dua atribut, yaitu nama dan NIM. Berbeda dengan contoh sebelumnya yang langsung memberi nilai di dalam kode, program ini meminta pengguna untuk mengisi data melalui input. Nama mahasiswa dibaca menggunakan `getline(cin, mhs.nama)` agar bisa menampung spasi, sedangkan NIM dimasukkan dengan `cin >> mhs.NIM`. Setelah itu, program menampilkan kembali data yang sudah dimasukkan ke layar. Program ini sederhana dan mudah dipahami sebagai contoh penggunaan struct dengan input dari user.

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

Guided 5

```

#include <iostream>
using namespace std;

using namespace std;

int main() {
    cout << "Nama : Bintang \nNIM : 103112400116" << endl;

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

```

Screenshots Output

```
Nama : Bintang
NIM : 103112400116
1
1
3
PS C:\semester3\struktur data\modul1> 
```

Deskripsi:

Program di atas menampilkan data sederhana berupa nama dan NIM, lalu memperlihatkan cara kerja operator increment pada variabel. Pertama, variabel nomor diberi nilai awal 1. Saat ditampilkan, urutannya adalah: 1 (nilai awal), 1 lagi (karena ditampilkan dulu baru ditambah dengan nomor++), dan 3 (karena ++nomor menambah nilai dulu baru menampilkannya). Program ini berguna untuk memahami perbedaan antara post-increment (nomor++) dan pre-increment (++nomor).

H. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

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

int main() {
    float a, b;
    cout << "Masukkan dua bilangan float: ";
    cin >> a >> b;

    cout << "Penjumlahan: " << a + b << endl;
    cout << "Pengurangan: " << a - b << endl;
    cout << "Perkalian : " << a * b << endl;
    if(b != 0)
        cout << "Pembagian : " << a / b << endl;
    else
        cout << "Pembagian tidak bisa (b=0)" << endl;

    return 0;
}
```

Screenshots Output

```
PS C:\semester3\struktur data\modul1> cd "c:\semester3\struktur data\modul1\" ; if
Masukkan dua bilangan float: 3 2
Penjumlahan: 5
Pengurangan: 1
Perkalian : 6
Pembagian : 1.5
PS C:\semester3\struktur data\modul1> []
```

Deskripsi:

Program ini menerima input dua bilangan bertipe float dari pengguna, lalu menghitung hasil penjumlahan, pengurangan, perkalian, dan pembagiannya. Pada operasi pembagian, program menggunakan percabangan if untuk mengecek apakah penyebut (b) sama dengan nol. Jika $b = 0$, program menampilkan pesan bahwa pembagian tidak bisa dilakukan, sehingga program aman dari error runtime.

I. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 2

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

string satuan[] =
{"nol","satu","dua","tiga","empat","lima","enam","tujuh","delapan","sembilan",
 "sepuluh","sebelas","dua belas","tiga belas","empat belas","lima belas",
 "enam belas","tujuh belas","delapan belas","sembilan belas"};

string puluhan[] = {"","","dua puluh","tiga puluh","empat puluh","lima puluh",
 "enam puluh","tujuh puluh","delapan puluh","sembilan puluh"};

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

    if (n < 20) cout << satuan[n];
    else if (n < 100) {
        cout << puluhan[n/10];
        if (n % 10 != 0) cout << " " << satuan[n%10];
    }
    else if (n == 100) cout << "seratus";
    else cout << "di luar jangkauan";

    return 0;
}
```


Screenshots Output

```
Masukkan angka (0-100): 79
tujuh puluh sembilan
PS C:\semester3\struktur data\modul1>
```

Deskripsi:

Program ini berfungsi untuk mengubah angka menjadi bentuk tulisan dalam bahasa Indonesia dengan batas antara 0 hingga 100. Angka-angka dasar dari nol sampai sembilan belas disimpan dalam array satuan, sedangkan angka kelipatan sepuluh dari dua puluh hingga sembilan puluh ada di array puluhan. Saat pengguna memasukkan angka, program akan memeriksa nilainya. Jika kurang dari 20, hasilnya langsung diambil dari array satuan. Jika berada di antara 20 sampai 99, program menampilkan nilai puluhannya, lalu menambahkan satuan jika angkanya tidak bulat. Untuk angka 100 akan ditampilkan tulisan "seratus", sedangkan angka di luar jangkauan akan diberi pesan khusus. Dengan logika sederhana ini, program dapat mengubah angka menjadi kata-kata secara otomatis.

- J. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 3

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

int main() {
    int n;
    cout << "Input: ";
    cin >> n;

    cout << "Output:" << endl;
    for(int i = n; i >= 1; i--) {
        for(int s = 0; s < n - i; s++) {
            cout << " ";
        }

        for(int j = i; j >= 1; j--) {
            cout << j << " ";
        }

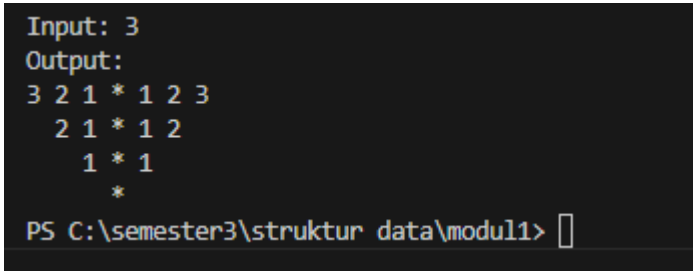
        cout << "* ";

        for(int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }

    for(int s = 0; s < n; s++) {
```

```
    cout << " ";  
}  
cout << "*" << endl;  
  
return 0;  
}
```

Screenshots Output



```
Input: 3  
Output:  
3 2 1 * 1 2 3  
 2 1 * 1 2  
  1 * 1  
   *  
PS C:\semester3\struktur data\modul1> 
```

Deskripsi:

Program di atas digunakan untuk menampilkan pola angka simetris dengan tambahan tanda bintang * di bagian tengahnya. Pertama, pengguna diminta memasukkan sebuah angka n sebagai tinggi pola. Setiap baris dicetak dengan susunan spasi di awal, kemudian deretan angka menurun dari i hingga 1, diikuti tanda *, lalu angka menaik dari 1 hingga i . Proses ini berulang dari baris pertama hingga baris terakhir. Setelah semua baris tercetak, program menambahkan satu baris akhir berupa spasi sejajar dengan tinggi pola dan sebuah tanda * di tengah. Hasilnya berupa pola angka berbentuk piramida yang simetris dan rapi.

K. Kesimpulan

Semua program yang sudah dibahas menunjukkan dasar-dasar penting dalam pemrograman C++ dengan contoh yang sederhana dan mudah dipahami. Mulai dari penggunaan percabangan (if-else dan switch) untuk pengambilan keputusan, perulangan (for, while, do-while) untuk mengulang proses, hingga struct untuk menyimpan dan mengelola data. Selain itu, ada juga contoh program yang mengubah angka menjadi teks dengan memanfaatkan array serta program yang mencetak pola angka menggunakan kombinasi logika perulangan dan spasi. Secara keseluruhan, program-program ini melatih pemahaman tentang logika dasar, struktur kontrol, manipulasi data, dan penyajian output di C++, yang merupakan fondasi utama sebelum masuk ke pemrograman tingkat lanjut.

L. Referensi

Jodi, M. (2020). Algoritma dan Struktur data.

Ginting, S. H. N., Effendi, H., Kumar, S., Marsisno, W., Sitanggang, Y. R. U., Anwar, K., ... & Smrti, N. N. E. (2024). Pengantar struktur data. *Penerbit Mifandi Mandiri Digital*, 1(01).

Ritonga, A., & Yahfizham, Y. (2023). Studi literatur perbandingan bahasa pemrograman C++ dan bahasa pemrograman Python pada algoritma pemrograman. *Jurnal Teknik Informatika dan Teknologi Informasi (JUTITI)*, 3(3).