

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL 3  
PENGENALAN CODE BLOCKS**



**Disusun Oleh :**

**NAMA : MUHAMMAD ATHA RIZQI  
RADITYA  
NIM : 103112400107**

**Dosen**

**WAHYU ANDI SAPUTRA**

**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>
#include <iomanip>
using namespace std;

struct Mahasiswa {
    string nama;
    string nim;
    float uts;
    float uas;
    float tugas;
    float nilaiAkhir;
};

float hitungNilaiAkhir(float uts, float uas, float tugas) {
    return 0.3 * uts + 0.4 * uas + 0.3 * tugas;
}

int main() {
    Mahasiswa data[10];
    int jumlah = 0;
    char lagi;

    cout << "=== Program Input Data Mahasiswa ===" << endl;

    do {
        cout << "\nData ke-" << jumlah + 1 << endl;
        cout << "Nama   : ";
        cin.ignore();
        getline(cin, data[jumlah].nama);
        cout << "NIM    : ";
        getline(cin, data[jumlah].nim);
        cout << "Nilai UTS : ";
        cin >> data[jumlah].uts;
        cout << "Nilai UAS : ";
        cin >> data[jumlah].uas;
```

```

        cout << "Nilai Tugas : ";
        cin >> data[jumlah].tugas;

        data[jumlah].nilaiAkhir = hitungNilaiAkhir(data[jumlah].uts, data[jumlah].uas,
data[jumlah].tugas);

        jumlah++;

        if (jumlah >= 10) {
            cout << "\nData sudah mencapai batas maksimal (10 mahasiswa)." << endl;
            break;
        }

        cout << "Tambah data lagi? (y/t): ";
        cin >> lagi;
    } while (lagi == 'y' || lagi == 'Y');

    cout << "\n=== Daftar Nilai Mahasiswa ===" << endl;
    cout << left << setw(5) << "No"
        << setw(20) << "Nama"
        << setw(15) << "NIM"
        << setw(8) << "UTS"
        << setw(8) << "UAS"
        << setw(8) << "Tugas"
        << setw(12) << "Nilai Akhir" << endl;
    cout << string(75, '-') << endl;

    for (int i = 0; i < jumlah; i++) {
        cout << left << setw(5) << i + 1
            << setw(20) << data[i].nama
            << setw(15) << data[i].nim
            << setw(8) << fixed << setprecision(2) << data[i].uts
            << setw(8) << data[i].uas
            << setw(8) << data[i].tugas
            << setw(12) << data[i].nilaiAkhir << endl;
    }

    cout << "\nProgram selesai." << endl;
    return 0;
}

```

Screenshots Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\ASUS\Documents\Struktur Data> cd "c:\Users\ASUS\Documents\Struktur Data\" ; if ($?) { g++ modul3s1.cpp -o
• === Program Input Data Mahasiswa ===

Data ke-1
Nilai UAS : 90
Nilai UAS : 90
Nilai Tugas : 95
Tambah data lagi? (y/t): t

=== Daftar Nilai Mahasiswa ===
No Nama NIM UTS UAS Tugas Nilai Akhir
-----
1 99 50.00 90.00 95.00 79.50

Program selesai.
```

### Deskripsi:

Program ini berfungsi untuk menyimpan dan menampilkan data maksimal sepuluh mahasiswa, di mana setiap mahasiswa memiliki atribut nama, NIM, nilai UTS, UAS, tugas, dan nilai akhir. Nilai akhir dihitung menggunakan fungsi `hitungNilaiAkhir()` dengan rumus  $0.3 \times \text{UTS} + 0.4 \times \text{UAS} + 0.3 \times \text{Tugas}$ . Program menggunakan struktur Mahasiswa untuk menampung data setiap mahasiswa, kemudian menginput data melalui perulangan `do-while` yang memungkinkan pengguna menambah data hingga batas maksimum. Setelah semua data dimasukkan, program menampilkan daftar nilai mahasiswa dalam bentuk tabel yang rapi menggunakan manipulasi output `setw()` dan `setprecision()`, lalu menutup dengan pesan bahwa program telah selesai dijalankan.

### Guided 2

h

```
#ifndef PELAJARAN_H
#define PELAJARAN_H
#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran create_pelajaran(string namaMapel, string kodeMapel);
void tampil_pelajaran(pelajaran pel);

#endif
```

cpp

```
#include <iostream>
```

```
#include "pelajaran.h"

using namespace std;

pelajaran create_pelajaran(string namaMapel, string kodeMapel) {
    pelajaran p;
    p.namaMapel = namaMapel;
    p.kodeMapel = kodeMapel;
    return p;
}

void tampil_pelajaran(pelajaran pel) {
    cout << "nama pelajaran : " << pel.namaMapel << endl;
    cout << "nilai : " << pel.kodeMapel << endl;
}
```

main

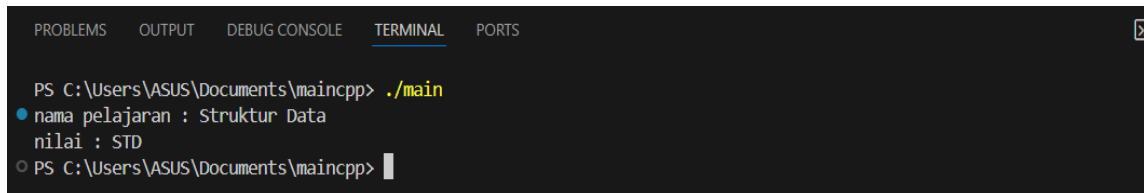
```
#include <iostream>
#include "pelajaran.h"
using namespace std;

int main() {
    string namapel = "Struktur Data";
    string kodepel = "STD";

    pelajaran pel = create_pelajaran(namelapel, kodepel);
    tampil_pelajaran(pel);

    return 0;
}
```

## Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS\Documents\maincpp> ./main
nama pelajaran : Struktur Data
nilai : STD
PS C:\Users\ASUS\Documents\maincpp>
```

## Deskripsi:

Program ini dibuat menggunakan bahasa C++ untuk mendemonstrasikan konsep Abstract Data Type (ADT) melalui struktur pelajaran yang memiliki dua atribut, yaitu namaMapel dan kodeMapel. Program terdiri dari tiga file — pelajaran.h, pelajaran.cpp, dan main.cpp — yang memisahkan antara deklarasi, implementasi, dan eksekusi. Fungsi `create_pelajaran()` digunakan untuk membuat objek pelajaran dengan nilai tertentu, sedangkan `tampil_pelajaran()` menampilkan isi objek tersebut ke layar. Secara keseluruhan, program ini menunjukkan cara mendefinisikan, mengimplementasikan, dan menggunakan ADT dalam bentuk modular yang terstruktur.

## Guided 3

```
#include <iostream>

using namespace std;

void tampilArray(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << "t";
        }
        cout << endl;
    }
}

void tukarArray(int arr1[3][3], int arr2[3][3], int baris, int kolom) {
    int temp = arr1[baris][kolom];
    arr1[baris][kolom] = arr2[baris][kolom];
    arr2[baris][kolom] = temp;
}
```

```
}
```

```
void tukarPointer(int *ptr1, int *ptr2) {
```

```
    int temp = *ptr1;
```

```
    *ptr1 = *ptr2;
```

```
    *ptr2 = temp;
```

```
}
```

```
int main() {
```

```
    int A[3][3] = {
```

```
        {1, 2, 3},
```

```
        {4, 5, 6},
```

```
        {7, 8, 9}
```

```
    };
```

```
    int B[3][3] = {
```

```
        {9, 8, 7},
```

```
        {6, 5, 4},
```

```
        {3, 2, 1}
```

```
    };
```

```
    int x = 10, y = 20;
```

```
    int *p1 = &x;
```

```
    int *p2 = &y;
```

```
    cout << "=== Isi Array A ===" << endl;
```

```
    tampilArray(A);
```

```
    cout << "\n=== Isi Array B ===" << endl;
```

```
    tampilArray(B);
```

```
cout << "\nMenukar A[1][2] dengan B[1][2]..." << endl;
tukarArray(A, B, 1, 2);

cout << "\n=== Array A setelah ditukar ===" << endl;
tampilArray(A);
cout << "\n=== Array B setelah ditukar ===" << endl;
tampilArray(B);

cout << "\n=== Sebelum penukaran pointer ===" << endl;
cout << "x = " << x << ", y = " << y << endl;

tukarPointer(p1, p2);

cout << "\n=== Setelah penukaran pointer ===" << endl;
cout << "x = " << x << ", y = " << y << endl;

return 0;
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS\Documents\maincpp> cd "c:\Users\ASUS\Documents\maincpp\" ; if ($?) { g++ modul3s3.cpp -o modul3s3 } ;
=== Isi Array A ===
1 2 3
4 5 6
7 8 9

=== Isi Array B ===
9 8 7
6 5 4
3 2 1

Menukar A[1][2] dengan B[1][2]...

=== Array A setelah ditukar ===
1 2 3
4 5 4
7 8 9

=== Array B setelah ditukar ===
9 8 7
6 5 6
3 2 1

=== Sebelum penukaran pointer ===
x = 10, y = 20

=== Setelah penukaran pointer ===
x = 20, y = 10
```

#### Deskripsi:

Program ini merupakan implementasi konsep dasar struktur data menggunakan array 2 dimensi dan pointer dalam bahasa C++. Di dalamnya terdapat dua array 3×3 (A dan B) serta dua pointer integer yang masing-masing menunjuk ke variabel x dan y. Program memiliki beberapa fungsi: tampilArray() untuk menampilkan isi array 2D, tukarArray() untuk menukar elemen tertentu antara dua array pada posisi yang sama, dan tukarPointer() untuk menukar nilai dua variabel melalui pointer. Melalui proses ini, program menunjukkan bagaimana fungsi, array 2D, dan pointer dapat bekerja bersama untuk memanipulasi data secara efisien dan terstruktur.

#### D. Kesimpulan

Kesimpulan dari keempat program C++ tersebut adalah bahwa masing-masing program mengajarkan konsep dasar pemrograman dan logika perhitungan menggunakan array, pointer, fungsi, serta struktur perulangan. Dari program pertama, dipelajari cara mengolah data mahasiswa, menghitung rata-rata, dan menampilkan hasil dalam format tabel. Program kedua memperkenalkan array dinamis dan penggunaan pointer untuk mengakses serta memanipulasi data. Program ketiga menekankan penggunaan fungsi terpisah (modularisasi) untuk menghitung rata-rata, nilai maksimum, dan minimum dari sekumpulan data. Sementara program keempat mengajarkan konsep perulangan bersarang untuk mencetak pola angka berbentuk segitiga. Secara keseluruhan, keempat program ini membantu memahami logika dasar, manipulasi data, penggunaan fungsi, array, pointer, serta struktur kontrol dalam bahasa pemrograman C++.

#### E. Referensi

- Alhazmi, A., & Huang, W. (2020). Teaching loops concept through visualization construction. *International Journal of Instruction*. 13(4), 399-114.

- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2020). Dampak penggunaan bahasa pemrograman terhadap pemahaman konsep loop pada siswa K-12. *Pendidikan dan Teknologi Informasi*, 25(1), 987-1000.
- Stroustrup, B. (1999). Tinjauan umum bahasa pemrograman C++. *ACM SIGPLAN Notices*, 34(4), 7-18.
- Stroustrup, B. (1999). Tinjauan umum bahasa pemrograman C++. *ACM SIGPLAN Notices*, 34(4), 7-18.