

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

Almas Fa'iq Khairul Ikhwan
21104029

Dosen

Diah Septiani S.Kom M.Cs

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Pengajaran pemrograman tingkat dasar merupakan sebuah bidang yang telah diteliti secara ekstensif, mencakup berbagai aspek mulai dari desain kurikulum, pemilihan bahasa pemrograman, hingga pendekatan pedagogis yang efektif. Memahami lanskap riset ini memberikan fondasi berbasis bukti yang kuat bagi para pengajar untuk merancang pengalaman belajar yang optimal bagi mahasiswa pemula [1].

Dua pilar utama dalam menstrukturkan program adalah *Abstract Data Type* (ADT) dan Fungsi. ADT, yang dalam C++ diimplementasikan melalui *struct*, memungkinkan pengelompokan data terkait menjadi satu unit yang logis. Sementara itu, fungsi memungkinkan pemecahan masalah kompleks menjadi sub-program yang lebih kecil dan dapat digunakan kembali. Kedua konsep ini merupakan alat fundamental untuk mengelola kompleksitas, meskipun penguasaannya sering menjadi tantangan utama bagi mahasiswa [2].

Inti dari logika pemrograman terletak pada penggunaan struktur kontrol seperti percabangan (*conditional*). Kemampuan untuk mengarahkan alur eksekusi program menggunakan pernyataan *if-else* adalah prasyarat fundamental untuk dapat menulis kode yang fungsional. Riset secara konsisten menunjukkan bahwa transisi dari pemahaman sintak ke penerapan logika percabangan untuk memecahkan masalah adalah salah satu rintangan terbesar yang dihadapi oleh pemrogram pemula [2].

B. Guided 1

```
#include <iostream>
#include <conio.h> // untuk fungsi getch()

using namespace std;

int main()
{
    int x, y; // variabel
    int *px; // pointer
    x = 87; // x menampung nilai 87
    px = &x; // pointer px menyimpan alamat dari x
    y = *px; // y menampung nilai yang ditunjuk oleh px (nilai tidak terdefinisi karena px belum
    diinisialisasi)

    cout << "Alamat x = " << &x << endl; // & adalah operator address of
    cout << "Isi px = " << px << endl; // px menyimpan alamat dari x
    cout << "Isi x = " << x << endl; // x menampilkan nilai 87
    cout << "Nilai px = " << *px << endl; // * adalah operator dereference
    cout << "Nilai y = " << y << endl; // y menampilkan nilai 87

    getch(); // fungsi untuk menahan layar
    return 0;
}
```

Screenshots Output

```
Alamat x = 0x5ffe70  
Isi px = 0x5ffe70  
Isi x = 87  
Nilai px = 87  
Nilai y = 87  
□
```

Deskripsi:

Program ini mendemonstrasikan konsep dasar *pointer* di C++. *Pointer* px digunakan untuk menyimpan alamat memori dari variabel x, kemudian nilai x diakses secara tidak langsung (*dereferencing*) untuk disalin ke variabel y. *Output* program membuktikan bahwa alamat x dan isi px sama, serta nilai x dan y juga menjadi sama.

Guided 2

```
#include <iostream>
#include <conio.h>
#define MAX 5 // Ukuran array ditetapkan sebagai 5

using namespace std;

int main()
{
    int i, j;
    float nilai[MAX]; // Array 1 dimensi
    static int nilai_tahun[MAX][MAX] = { // Array 2 dimensi (5x5)
        {0, 2, 2, 0, 0},
        {0, 1, 1, 1, 0},
        {0, 3, 3, 3, 0},
        {4, 4, 0, 0, 4},
        {5, 0, 0, 0, 5}};

    // Input data array 1 dimensi
    for (i = 0; i < MAX; i++)
    {
        cout << "Masukkan nilai ke-" << i + 1 << " ";
        cin >> nilai[i];
    }

    // Menampilkan isi array 1 dimensi
    cout << "\nData nilai siswa:\n";
    for (i = 0; i < MAX; i++)
    {
        cout << "Nilai ke-" << i + 1 << " = " << nilai[i] << endl;
    }

    // Menampilkan isi array 2 dimensi
    cout << "\nNilai tahunan:\n";
    for (i = 0; i < MAX; i++)
    {
        for (j = 0; j < MAX; j++)
        {
            cout << nilai_tahun[i][j] << " ";
        }
        cout << endl; // Tambahkan baris baru setelah menampilkan array 2D
    }
    getch(); // fungsi untuk menahan layar
    return 0;
}
```

Screenshots Output

```
Masukkan nilai ke-1 85
Masukkan nilai ke-2 90
Masukkan nilai ke-3 95
Masukkan nilai ke-4 100
Masukkan nilai ke-5 95

Data nilai siswa:
Nilai ke-1 = 85
Nilai ke-2 = 90
Nilai ke-3 = 95
Nilai ke-4 = 100
Nilai ke-5 = 95

Nilai tahunan:
0 2 2 0 0
0 1 1 1 0
0 3 3 3 0
4 4 0 0 4
5 0 0 0 5
```

Deskripsi:

Program ini menunjukkan cara kerja *array* satu dimensi dan dua dimensi. *Array* satu dimensi diisi melalui *input* pengguna dan ditampilkan kembali, sedangkan *array* dua dimensi yang nilainya sudah ada di dalam kode ditampilkan dalam format matriks menggunakan perulangan bersarang (*nested loop*).

C. Unguided 1

```
/*
 * Nama      : Almas Fa'iq Khairul Ikhwan
 * NIM       : 21104029
 * Dosen     : Diah Septiani S.Kom M.Cs
 * Deskripsi  : Program ini menghitung nilai akhir dan indeks mahasiswa untuk mata kuliah Struktur
               Data berdasarkan 4 nilai CL0.
 */

#include <iostream>
#include <string>
#include <vector>
#include <conio.h>

// Soal 1: Deklarasi tipe bentukan 'nilaiSTD' menggunakan struct
struct nilaiSTD
{
    float CL01, CL02, CL03, CL04;
    float nilai_akhir;
    std::string indeks;
};

// Soal 6: Function untuk menghitung nilai akhir
float hitungNilaiAkhir(float c1, float c2, float c3, float c4)
{
    // Rumus: 30% CL01 + 30% CL02 + 20% CL03 + 20% CL04
    return (0.30f * c1) + (0.30f * c2) + (0.20f * c3) + (0.20f * c4);
}

// Soal 7: Function untuk menentukan indeks nilai
std::string tentukanIndeks(float nilai)
{
    if (nilai > 80)
    {
        return "A";
    }
    else if (nilai > 70)
    { // nilai <= 80
        return "AB";
    }
    else if (nilai > 65)
    { // nilai <= 70
        return "B";
    }
    else if (nilai > 60)
    { // nilai <= 65
        return "BC";
    }
    else if (nilai > 50)
    { // nilai <= 60
        return "C";
    }
    else if (nilai > 40)
    { // nilai <= 50
        return "D";
    }
    else
    { // nilai <= 40
        return "E";
    }
}

// Soal 4 & 5: Function untuk membuat dan mengisi struct nilaiSTD
int main()
{
    std::cout << "-----" << std::endl;
    std::cout << "Nama      : Almas Fa'iq Khairul Ikhwan" << std::endl;
    std::cout << "NIM       : 21104029" << std::endl;
    std::cout << "Tugas    : Laporan Praktikum Modul 1" << std::endl;
    std::cout << "-----\n" << std::endl;

    // Soal 2: Deklarasi variabel
    nilaiSTD data_mhs1;
    float in_clo1, in_clo2, in_clo3, in_clo4;

    // Soal 3: Meminta input dari user
    std::cout << "Masukkan nilai CL0-1: ";
    std::cin >> in_clo1;
    std::cout << "Masukkan nilai CL0-2: ";
    std::cin >> in_clo2;
    std::cout << "Masukkan nilai CL0-3: ";
    std::cin >> in_clo3;
    std::cout << "Masukkan nilai CL0-4: ";
    std::cin >> in_clo4;

    data_mhs1.CL01 = in_clo1;
    data_mhs1.CL02 = in_clo2;
    data_mhs1.CL03 = in_clo3;
    data_mhs1.CL04 = in_clo4;

    data_mhs1.nilai_akhir = hitungNilaiAkhir(data_mhs1.CL01, data_mhs1.CL02, data_mhs1.CL03,
    data_mhs1.CL04);
    data_mhs1.indeks = tentukanIndeks(data_mhs1.nilai_akhir);

    std::cout << "\n--- HASIL PERHITUNGAN ---" << std::endl;
    std::cout << "Nilai Akhir   : " << data_mhs1.nilai_akhir << std::endl;
    std::cout << "Indeks Nilai  : " << data_mhs1.indeks << std::endl;

    getch();
    return 0;
}
```

Screenshots Output

```
-----  
Nama    : Almas Fa'iq Khairul Ikhwan  
NIM     : 21104029  
Tugas   : Laporan Praktikum Modul 1  
-----  
  
Masukkan nilai CLO-1: 85  
Masukkan nilai CLO-2: 90  
Masukkan nilai CLO-3: 100  
Masukkan nilai CLO-4: 95  
  
--- HASIL PERHITUNGAN ---  
Nilai Akhir   : 91.5  
Indeks Nilai  : A
```

Deskripsi:

Kode C++ di atas dirancang untuk menyelesaikan serangkaian tugas dari soal 1 hingga 7.

1. Struct nilaiSTD (Soal 1): Didefinisikan sebuah struct untuk membungkus semua data nilai mahasiswa ke dalam satu tipe data bentukan.
2. Fungsi hitungNilaiAkhir (Soal 6): Fungsi ini menerima empat nilai CLO, menghitung nilai akhir sesuai bobot yang diberikan, dan mengembalikannya sebagai float.
3. Fungsi tentukanIndeks (Soal 7): Fungsi ini menerima satu nilai akhir dan menggunakan struktur percabangan if-else if untuk menentukan dan mengembalikan indeks nilai dalam bentuk string.
4. Fungsi main (Soal 2, 3, 4, 5): Fungsi utama ini menjadi pusat program. Di dalamnya, variabel yang dibutuhkan dideklarasikan (Soal 2). Program kemudian meminta input empat nilai CLO dari pengguna (Soal 3). Nilai-nilai ini kemudian diproses dengan memanggil fungsi hitungNilaiAkhir dan tentukanIndeks, dan hasilnya disimpan ke dalam variabel data_mhs1 yang bertipe struct nilaiSTD (pemenuhan Soal 4 & 5). Terakhir, program menampilkan hasil perhitungan nilai akhir dan indeks nya.

D. Kesimpulan

Praktikum ini berhasil mengimplementasikan konsep-konsep fundamental pemrograman dalam bahasa C++. Dengan menggunakan struct sebagai representasi Abstract Data Type (ADT), data dapat dikelola secara lebih terstruktur. Pemecahan masalah menjadi fungsi-fungsi terpisah seperti hitungNilaiAkhir dan tentukanIndeks membuktikan efektivitas pendekatan modular dalam membuat program yang bersih dan mudah dipahami. Logika percabangan juga berhasil diterapkan untuk menangani proses pengambilan keputusan berdasarkan rentang nilai.

E. Referensi

- [1] A. Pears *et al.*, “A survey of literature on the teaching of introductory programming,” *ACM SIGCSE Bulletin*, vol. 39, no. 4, pp. 204–223, Dec. 2007, doi: 10.1145/1345375.1345441.
- [2] A. Luxton-Reilly *et al.*, “Introductory Programming: A Systematic Literature Review,” 2018. doi: 10.1145/3293881.3295779.