

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

Geusan Edurais Aria Daffa
103122400026

Dosen

Diah Septiani, S.Kom., M.C.S

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

A. Dasar Teori

- Tipe Data Abstrak (ADT): Ini adalah konsep atau blueprint buat sebuah data. Fokusnya ke apa aja yang bisa dilakuin sama data itu, bukan gimana cara bikinnya.
- Record / Struct: Ini adalah implementasi dari ADT. Gunanya buat ngelompokkin beberapa data yang beda tipe (misal: angka dan tulisan) jadi satu wadah yang rapi, contohnya kayak nilaiSTD di soal.
- Pemrograman Modular: Ini prinsip buat mecah program gede jadi bagian-bagian kecil yang independen, yang disebut function. Tujuannya biar kode gampang dibaca, dipake ulang, dan diperbaiki.
- Percabangan (IF-ELSE): Ini adalah cara program buat ngambil keputusan. Program bakal ngecek sebuah kondisi, terus nentuin perintah mana yang mau dijalankan.

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

Guided 1

```
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    int x, y;
    int *px;
    x = 87;
    px = &x;
    y = *px;

    cout << "Alamat x = " << &x << endl;
    cout << "Isi px = " << &px << endl;
    cout << "isi x = " << x << endl;
    cout << "Nilai px = " << *px << endl;
    cout << "Nilai y = " << y << endl;

    getch();
    return 0;
}
```

Screenshots Output

```
$ ./"index.exe"
Alamat x = 0x3291bff988
Isi px = 0x3291bff980
isi x = 87
Nilai px = 87
Nilai y = 87
```

Deskripsi:

kode itu intinya nunjukin gimana sebuah pointer bernama px bisa nyimpen alamat memori dari variabel x. Setelah px tau alamatnya si x, variabel y kemudian ngambil nilainya bukan dari x langsung, tapi dari alamat yang disimpan sama px itu. Proses ini pake tanda * (bintang), makanya hasil akhirnya nunjukin kalo nilai si x, si y, dan nilai yang diliat lewat *px itu semuanya jadi sama, yaitu 87.

Guided 2

```
#include <iostream>
#include <conio.h>
#define MAX 5

using namespace std;

int main()
{
    int i, j;
    float nilai[MAX];
    static int nilai_tahun[MAX][MAX] = {
        {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},
    };

    for (i = 0; i < MAX; i++)
    {
        cout << "Masukkan nilai ke-" << i + 1 << " : ";
        cin >> nilai[i];
    }

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

    cout << "\nNilai tahunan:\n";
    for (i = 0; i < MAX; i++)
    {
        for (j = 0; j < MAX; j++)
        {
            cout << nilai_tahun[i][j] << " ";
        }
        cout << endl;
    }
}
```

```
}

getch(); // Menahan Layar
return 0;
}
```

Screenshots Output

```
$ ./"contoh3.exe"
Masukkan nilai ke-1:60
Masukkan nilai ke-2:70
Masukkan nilai ke-3:80
Masukkan nilai ke-4:90
Masukkan nilai ke-5:100

Data nilai siswa:
Nilai ke-1 = 60
Nilai ke-2 = 70
Nilai ke-3 = 80
Nilai ke-4 = 90
Nilai ke-5 = 100

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:

Kode itu intinya nunjukin cara pake dua jenis array: yang pertama, ada array satu dimensi bernama nilai yang awalnya kosong, terus programnya minta kamu buat ngisi lima angka ke dalemnya. Setelah keisi, semua angka yang dimasukin tadi ditampilin lagi ke cmd. Selain itu, ada juga array dua dimensi bernama nilai_tahun yang datanya udah disiapin dari awal di dalem kode, dan di bagian akhir programnya cuma nampilin semua isi tabel itu secara rapi baris per baris.

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

Unguided

```
1. CLO1          : Int
CLO2             : Int
CLO3             : Int
CLO4             : Int
nilai_akhir      : Int
indeks_nilai     : STRING
```

2. **DECLARE** mahasiswa, mahasiswi: **nilaidata**

3. **Program HitungNilai**

Kamus

CLO1, CLO2, CLO3, CLO4: Int

Algoritma

```
    write("Masukkan nilai CLO1: ")
    input(CLO1)
        write("Masukkan nilai CLO2: ")
    input(CLO2)
        write("Masukkan nilai CLO3: ")
    input(CLO3)
        write("Masukkan nilai CLO4: ")
    input(CLO4)

    output("Nilai dari CLO1: ", CLO1)
        output("Nilai dari CLO2: ", CLO2)
        output("Nilai dari CLO3: ", CLO3)
    output("Nilai dari CLO4: ", CLO4)
```

4. Function ProsesNilaiCLO(newCLO1: integer, newCLO2: integer, newCLO3: integer, newCLO4: integer) : **nilaidata**

5. Function ProsesNilaiCLO(newCLO1: integer, newCLO2: integer, newCLO3: integer, newCLO4: integer) : **nilaidata**

KAMUS

hasilNilai: nilaidata

Algoritma

hasilNilai.CLO1 <- newCLO1

hasilNilai.CLO2 <- newCLO2

hasilNilai.CLO3 <- newCLO3

hasilNilai.CLO4 <- newCLO4

hasilNilai.nilai_akhir <- (0.30 * hasilNilai.CLO1) + (0.30 * hasilNilai.CLO2) + (0.20 * hasilNilai.CLO3) + (0.20 * hasil.CLO4)

if (hasilNilai.nilai_akhir >= 80) THEN

 hasilNilai.indeks_nilai <- "A"

else if (hasilNilai.nilai_akhir >= 75) THEN

 hasilNilai.indeks_nilai <- "AB"

else if (hasilNilai.nilai_akhir >= 70) THEN

 hasilNilai.indeks_nilai <- "B"

else if (hasilNilai.nilai_akhir >= 65) THEN

 hasilNilai.indeks_nilai <- "BC"

else if (hasilNilai.nilai_akhir >= 60) THEN

 hasilNilai.indeks_nilai <- "C"

else if (hasilNilai.nilai_akhir >= 50) THEN

 hasilNilai.indeks_nilai <- "D"

else

 hasilNilai.indeks_nilai <- "E"

end if

return hasil

6. function HitungNilaiAkhir(clo1: integer, clo2: integer, clo3: integer, clo4: integer) -> integer

KAMUS

akhir: integer

Algoritma

akhir <- (0.30 * clo1) + (0.30 * clo2) + (0.20 * clo3) + (0.20 * clo4)

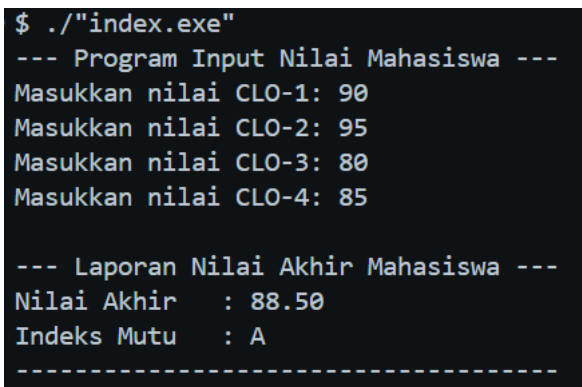
return nilaiAkhir

7. function TentukanIndeks(nilaiAkhir: int) -> string

Algoritma

```
If nilaiAkhir > 80 then
    Return "A"
Else if nilaiAkhir > 70 then
    return "AB"
Else if nilaiAkhir > 65 then
    Return "B"
Else if nilaiAkhir > 60 then
    Return "BC"
Else if nilaiAkhir > 50 then
    Return "C"
Else if nilaiAkhir > 40 then
    Return "D"
Else
    Return "E"
Endif
```

Screenshots Output



```
$ ./"index.exe"
--- Program Input Nilai Mahasiswa ---
Masukkan nilai CLO-1: 90
Masukkan nilai CLO-2: 95
Masukkan nilai CLO-3: 80
Masukkan nilai CLO-4: 85

--- Laporan Nilai Akhir Mahasiswa ---
Nilai Akhir    : 88.50
Indeks Mutu    : A
-----
```

Deskripsi:

Semua Kode tersebut dimulai dari mendefinisikan tipe data variable buat nyimpen data nilai, bikin program buat input data, sampe ngerancang fungsi-fungsi buat ngolah data nilai itu jadi nilai akhir dan char huruf/nilainya.

E. Kesimpulan

Kesimpulan dari laprak ini menunjukkan bagaimana konsep fundamental seperti Tipe Data Abstrak (Struct) untuk mengorganisir data, fungsi untuk modularitas, dan percabangan (IF-ELSE) untuk pengambilan keputusan, secara bersama-sama digunakan untuk membangun sebuah program yang terstruktur dan logis.

F. Referensi

Lafore, R. (2002). *Data Structures and Algorithms in C++*. Sams Publishing.