

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Davis Arvaputra Dwiansyah
NIM : 103122400034

Dosen

Diah Septiani S.Kom M.Cs

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

A. Dasar Teori

C++ adalah bahasa pemrograman multi-paradigma yang mendukung pemrograman prosedural, berorientasi objek, dan generik. Bahasa ini memberikan fleksibilitas serta efisiensi tinggi, karena dikompilasi dan memiliki sistem tipe statis, sehingga mampu mengontrol penggunaan memori dengan baik [1].

Sebagai ekstensi dari bahasa C, C++ terus dikembangkan dengan fitur modern yang mendukung skalabilitas, paralelisme, dan performa. Oleh karena itu, C++ banyak digunakan pada pengembangan aplikasi dengan kebutuhan komputasi tinggi, seperti simulasi ilmiah, sistem perangkat keras, hingga komputasi paralel modern [2].

B. Guided

Guided 1

```
#include <iostream>

using namespace std;

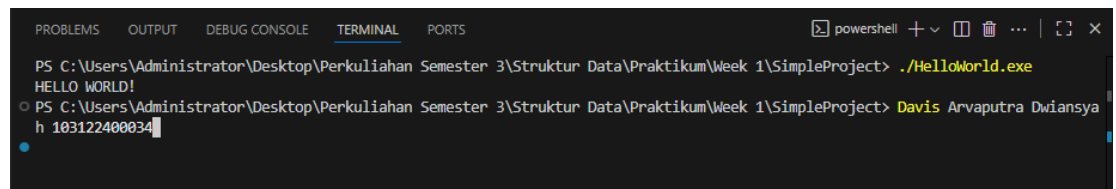
int main() {

    cout<<"HELLO WORLD!"<<endl;

    return 0;

}
```

Screenshots Output:



```
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject> ./HelloWorld.exe
HELLO WORLD!
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject> Davis Arvaputra Dwiansyah
h 10312240034
```

Deskripsi:

Program “Hello, World!” adalah tes fundamental untuk memastikan bahwa compiler dan lingkungan pengembangan C++ berjalan dengan benar. Program ini menggunakan pustaka <iostream> yang menyediakan std::cout untuk mengirimkan output ke konsol. Contoh ini sering dipakai sebagai

langkah awal untuk memperkenalkan konsep dasar stream I/O (Input/Output) serta struktur minimum program C++ [3].

Selain itu, penggunaan program sederhana seperti “Hello, World!” juga dianggap penting dalam pendidikan pemrograman karena membantu mahasiswa memahami dasar sintaksis bahasa dan memberikan umpan balik langsung terhadap keberhasilan kompilasi serta eksekusi. Dengan demikian, latihan ini bukan hanya sekadar percobaan teknis, melainkan juga strategi pedagogis yang efektif untuk meningkatkan pemahaman awal pemrograman [4].

Guided 2

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

using namespace std;

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

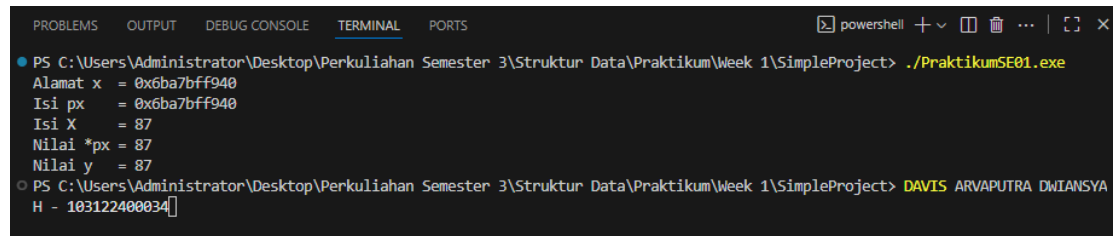
    cout<< "Alamat x = "<< &x <<endl;
    cout<< "Isi px  = "<< px <<endl;
    cout<< "Isi X   = "<< x <<endl;
    cout<< "Nilai *px = "<< *px <<endl;
    cout<< "Nilai y  = "<< y <<endl;

    getch();
}
```

```
return 0;
}
```

Screenshots

Output:

A screenshot of a terminal window showing the output of a C++ program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The output text is as follows:

```
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject> ./PraktikumSE01.exe
Alamat x = 0x6ba7bff940
Isi px = 0x6ba7bff940
Isi X = 87
Nilai *px = 87
Nilai y = 87
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject> DAVIS ARVAPUTRA DWIANSYA
H - 10312240034
```

Deskripsi:

Program di atas mendemonstrasikan penggunaan **pointer** dalam bahasa C++. Variabel **x** didefinisikan dengan nilai 87, kemudian alamat memorinya disimpan pada pointer **px** dengan sintaks **px = &x**; . Nilai yang ditunjuk pointer (***px**) kemudian disalin ke variabel **y**. Melalui keluaran program, dapat ditampilkan: alamat dari variabel **x**, isi dari pointer **px** (alamat yang ditunjuk), nilai dari **x**, dereferensi ***px**, serta nilai **y**. Program ini menunjukkan hubungan langsung antara variabel, alamat memori, dan pointer sebagai representasi lokasi data di memori [5].

Konsep pointer sendiri merupakan salah satu fitur penting C++ karena memberi akses langsung ke memori dan memungkinkan manipulasi data secara efisien. Pointer tidak hanya mendukung pengelolaan memori dinamis, tetapi juga menjadi dasar dalam implementasi struktur data seperti linked list dan tree. Dalam konteks pembelajaran pemrograman, latihan sederhana seperti ini membantu mahasiswa memahami bagaimana alamat dan isi memori dapat diakses dan diubah melalui pointer [6].

Guided 3

```
#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}

    };

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

    // menampilkan 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

```

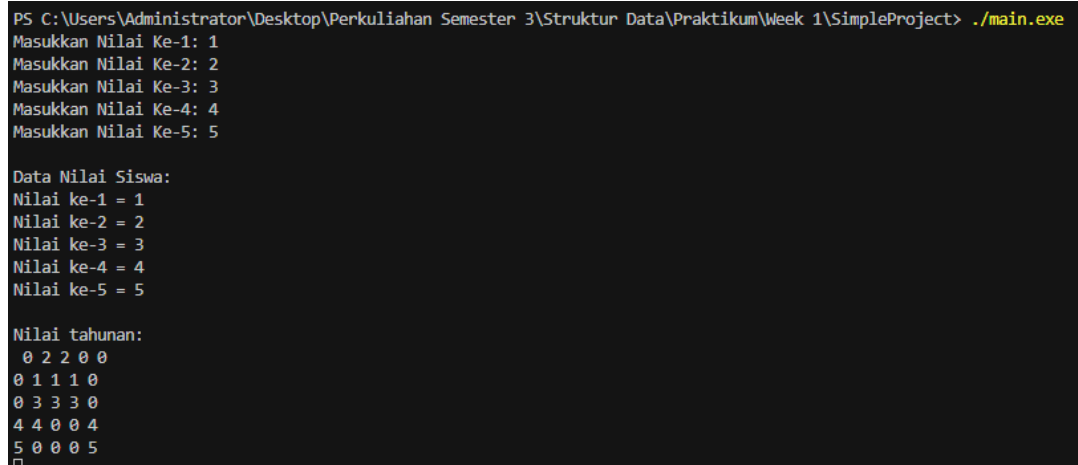
```

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

getch();
return 0;

```

Screenshots Output:



```

PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject> ./main.exe
Masukkan Nilai Ke-1: 1
Masukkan Nilai Ke-2: 2
Masukkan Nilai Ke-3: 3
Masukkan Nilai Ke-4: 4
Masukkan Nilai Ke-5: 5

Data Nilai Siswa:
Nilai ke-1 = 1
Nilai ke-2 = 2
Nilai ke-3 = 3
Nilai ke-4 = 4
Nilai ke-5 = 5

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 mendemonstrasikan penggunaan array satu dimensi dan dua dimensi di C++. Pertama, array nilai berukuran tetap (MAX = 5) digunakan untuk menyimpan sekumpulan data bertipe sama yang dimasukkan pengguna melalui cin, kemudian ditampilkan kembali dengan perulangan. Selanjutnya, program menampilkan elemen dari array dua dimensi nilai_tahun, yang merepresentasikan data dalam bentuk tabel. Latihan ini

memperlihatkan konsep dasar struktur data array serta mekanisme input-output konsol di C++, yang memang menjadi materi fundamental dalam pembelajaran pemrograman dasar [7].

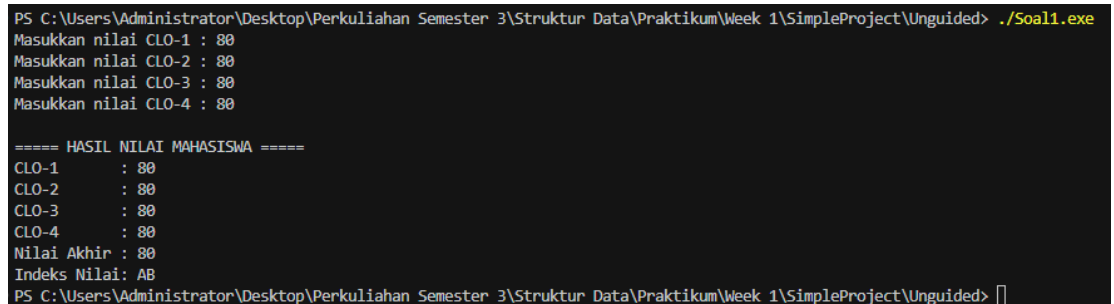
- C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```
TYPE nilaiSTD = RECORD
    CLO1 : REAL
    CLO2 : REAL
    CLO3 : REAL
    CLO4 : REAL
    nilaiAkhir : REAL
    indeks : STRING // berisi A, AB, B, BC, C, D, atau E
END RECORD
```

Screenshots

Output



```
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> ./Soal1.exe
Masukkan nilai CLO-1 : 80
Masukkan nilai CLO-2 : 80
Masukkan nilai CLO-3 : 80
Masukkan nilai CLO-4 : 80

===== HASIL NILAI MAHASISWA =====
CLO-1      : 80
CLO-2      : 80
CLO-3      : 80
CLO-4      : 80
Nilai Akhir : 80
Indeks Nilai: AB
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> █
```

Deskripsi:

Pseudocode TYPE nilaiSTD : RECORD mendefinisikan sebuah tipe data bentukan bernama nilaiSTD yang berfungsi untuk menyimpan data nilai mahasiswa secara terstruktur. Di dalam record ini terdapat empat atribut clo1, clo2, clo3, dan clo4 bertipe real untuk menyimpan nilai CLO 1 hingga CLO 4, satu atribut nilaiAkhir bertipe real untuk menampung hasil perhitungan akhir, serta atribut indeks bertipe string untuk menyimpan nilai mutu dalam bentuk huruf (A, AB, B, BC, C, D, atau E). Seluruh komponen

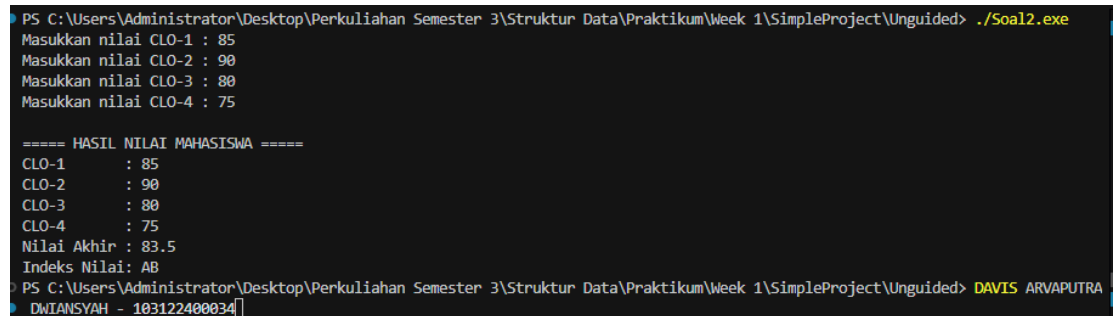
tersebut dikelompokkan dalam satu wadah sehingga data yang berhubungan dapat dikelola lebih terorganisir, dan definisi record diakhiri dengan END RECORD sebagai penutup struktur.

Unguided 2

```
TYPE nilaiSTD = RECORD
    CLO1 : REAL
    CLO2 : REAL
    CLO3 : REAL
    CLO4 : REAL
    nilaiAkhir : REAL
    indeks : STRING
END RECORD

DECLARE mahasiswa1 : nilaiSTD
DECLARE mahasiswa2 : nilaiSTD
```

Screenshots Output



```
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> ./Soal2.exe
Masukkan nilai CLO-1 : 85
Masukkan nilai CLO-2 : 90
Masukkan nilai CLO-3 : 80
Masukkan nilai CLO-4 : 75

===== HASIL NILAI MAHASISWA =====
CLO-1      : 85
CLO-2      : 90
CLO-3      : 80
CLO-4      : 75
Nilai Akhir : 83.5
Indeks Nilai: AB
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> DAVIS ARVAPUTRA
DWIANSYAH - 103122400034
```

Deskripsi:

Struktur nilaiSTD digunakan untuk menyimpan data nilai mahasiswa yang terdiri dari empat nilai CLO, satu nilai akhir, dan satu indeks huruf. Setelah struktur tersebut dibuat, dideklarasikan dua variabel yaitu mahasiswa1 dan mahasiswa2 yang masing-masing dapat menyimpan data lengkap seorang mahasiswa. Dengan cara ini, data yang berhubungan dapat dikelompokkan dalam satu wadah sehingga lebih teratur dan mudah digunakan.

Unguided 3

```
DECLARE CLO1, CLO2, CLO3, CLO4 : REAL

OUTPUT "Masukkan nilai CLO-1 : "
```


INPUT CLO1

OUTPUT "Masukkan nilai CLO-2 : "

INPUT CLO2

OUTPUT "Masukkan nilai CLO-3 : "

INPUT CLO3

OUTPUT "Masukkan nilai CLO-4 : "

INPUT CLO4

OUTPUT "Data nilai mahasiswa:"

OUTPUT "CLO-1 = ", CLO1

OUTPUT "CLO-2 = ", CLO2

OUTPUT "CLO-3 = ", CLO3

OUTPUT "CLO-4 = ", CLO4

Screenshots Output

```
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> g++ Soal3.cpp -o Soal3.exe
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> ./Soal3.exe
Masukkan nilai CLO-1 : 80
Masukkan nilai CLO-2 : 90
Masukkan nilai CLO-3 : 70
Masukkan nilai CLO-4 : 60

Data nilai mahasiswa:
CLO-1 = 80
CLO-2 = 90
CLO-3 = 70
CLO-4 = 60
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> DAVIS ARVAPUTRA
DWIANSYAH - 103122400034
```

Deskripsi:

Program ini meminta pengguna untuk memasukkan empat nilai, yaitu CLO1 sampai CLO4, kemudian menyimpannya dalam variabel bertipe real. Setelah semua nilai diinputkan, program menampilkan kembali hasil input tersebut ke layar. Dengan cara ini, data yang dimasukkan oleh pengguna dapat dicek ulang dan dipastikan sesuai sebelum diproses lebih lanjut.

Unguided 4

```
TYPE nilaiSTD = RECORD
    clo1 : REAL
    clo2 : REAL
    clo3 : REAL
```

```

    clo4 : REAL
    rataRata : REAL
END RECORD

FUNCTION hitungNilaiSTD(c1 : REAL, c2 : REAL, c3 : REAL, c4 : REAL) →
nilaiSTD
    DECLARE hasil : nilaiSTD
    SET hasil.clo1 ← c1
    SET hasil.clo2 ← c2
    SET hasil.clo3 ← c3
    SET hasil.clo4 ← c4
    SET hasil.rataRata ← (c1 + c2 + c3 + c4) / 4
    RETURN hasil
END FUNCTION

OUTPUT "Masukkan nilai CLO 1: "
INPUT n1
OUTPUT "Masukkan nilai CLO 2: "
INPUT n2
OUTPUT "Masukkan nilai CLO 3: "
INPUT n3
OUTPUT "Masukkan nilai CLO 4: "
INPUT n4

SET mahasiswa ← hitungNilaiSTD(n1, n2, n3, n4)

OUTPUT "Data Nilai Mahasiswa:"
OUTPUT "CLO 1: ", mahasiswa.clo1
OUTPUT "CLO 2: ", mahasiswa.clo2
OUTPUT "CLO 3: ", mahasiswa.clo3
OUTPUT "CLO 4: ", mahasiswa.clo4
OUTPUT "Rata-rata: ", mahasiswa.rataRata

```

Screenshots Output

```

PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> g++ Soal4.cpp -o Soal4.exe
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> ./Soal4.exe
Masukkan nilai CLO 1: 80
Masukkan nilai CLO 2: 70
Masukkan nilai CLO 3: 85
Masukkan nilai CLO 4: 90

Data Nilai Mahasiswa:
CLO 1: 80
CLO 2: 70
CLO 3: 85
CLO 4: 90
Rata-rata: 81.25
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> DAVIS ARVAPUTRA
DWIANSYAH - 103122400034

```

Deskripsi:

Algoritma ini mendefinisikan sebuah struktur nilaiSTD yang berisi empat atribut nilai CLO dan satu atribut rata-rata. Selanjutnya dibuat fungsi hitungNilaiSTD yang menerima empat nilai CLO, menyimpannya ke dalam struktur, lalu menghitung rata-rata dari keempat nilai tersebut. Di bagian utama program, pengguna diminta memasukkan nilai CLO1 sampai CLO4, kemudian fungsi dipanggil untuk mengolah input tersebut menjadi satu kesatuan data. Hasil akhirnya ditampilkan berupa nilai masing-masing CLO beserta rata-ratanya.

Unguided 5

```

TYPE nilaiSTD = RECORD
    clo1 : REAL
    clo2 : REAL
    clo3 : REAL
    clo4 : REAL
    rataRata : REAL
END RECORD

FUNCTION hitungNilaiSTD(c1 : REAL, c2 : REAL, c3 : REAL, c4 : REAL) → nilaiSTD
    DECLARE hasil : nilaiSTD
    SET hasil.clo1 ← c1
    SET hasil.clo2 ← c2
    SET hasil.clo3 ← c3
    SET hasil.clo4 ← c4
    SET hasil.rataRata ← (c1 + c2 + c3 + c4) / 4
    RETURN hasil
END FUNCTION

```

```

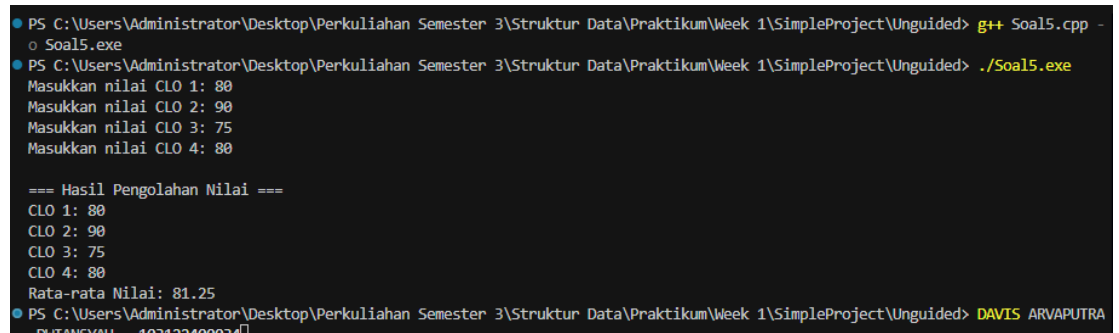
OUTPUT "Masukkan nilai CLO 1: "
INPUT n1
OUTPUT "Masukkan nilai CLO 2: "
INPUT n2
OUTPUT "Masukkan nilai CLO 3: "
INPUT n3
OUTPUT "Masukkan nilai CLO 4: "
INPUT n4

SET mahasiswa ← hitungNilaiSTD(n1, n2, n3, n4)

OUTPUT "=== Hasil Pengolahan Nilai ==="
OUTPUT "CLO 1: ", mahasiswa.clo1
OUTPUT "CLO 2: ", mahasiswa.clo2
OUTPUT "CLO 3: ", mahasiswa.clo3
OUTPUT "CLO 4: ", mahasiswa.clo4
OUTPUT "Rata-rata Nilai: ", mahasiswa.rataRata

```

Screenshots Output



```

PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> g++ Soal5.cpp -o Soal5.exe
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> ./Soal5.exe
Masukkan nilai CLO 1: 80
Masukkan nilai CLO 2: 90
Masukkan nilai CLO 3: 75
Masukkan nilai CLO 4: 80

=== Hasil Pengolahan Nilai ===
CLO 1: 80
CLO 2: 90
CLO 3: 75
CLO 4: 80
Rata-rata Nilai: 81.25
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided>

```

Deskripsi:

Algoritma ini menggunakan sebuah struktur nilaiSTD untuk menampung nilai CLO1 hingga CLO4 beserta rata-rata. Fungsi hitungNilaiSTD menerima empat nilai CLO sebagai input, menyimpannya ke dalam struktur, lalu menghitung rata-rata nilai tersebut. Pada bagian utama program, pengguna diminta mengisi nilai CLO1 sampai CLO4. Nilai-nilai tersebut diproses dengan memanggil fungsi, dan hasilnya ditampilkan berupa nilai tiap CLO beserta rata-ratanya. Dengan pendekatan ini, data dapat disimpan secara terstruktur dan hasil perhitungan rata-rata dapat langsung dilihat.

Unguided 6

```
TYPE nilaiSTD = RECORD
    clo1 : REAL
    clo2 : REAL
    clo3 : REAL
    clo4 : REAL
END RECORD

FUNCTION hitungNilaiAkhir(data : nilaiSTD) → REAL
    DECLARE hasil : REAL
    SET hasil ← (0.3 * data.clo1) + (0.3 * data.clo2) + (0.2 * data.clo3) + (0.2 *
data.clo4)
    RETURN hasil
END FUNCTION

DECLARE mahasiswa : nilaiSTD

OUTPUT "Masukkan nilai CLO 1: "
INPUT mahasiswa.clo1
OUTPUT "Masukkan nilai CLO 2: "
INPUT mahasiswa.clo2
OUTPUT "Masukkan nilai CLO 3: "
INPUT mahasiswa.clo3
OUTPUT "Masukkan nilai CLO 4: "
INPUT mahasiswa.clo4

SET nilaiAkhir ← hitungNilaiAkhir(mahasiswa)

OUTPUT "=== Hasil Nilai Akhir MK Struktur Data ==="
OUTPUT "Nilai Akhir: ", nilaiAkhir
```

Screenshots Output

```
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> g++ Soal6.cpp -
o Soal6.exe
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> ./Soal6.exe
Masukkan nilai CLO 1: 80
Masukkan nilai CLO 2: 90
Masukkan nilai CLO 3: 80
Masukkan nilai CLO 4: 90

=== Hasil Nilai Akhir MK Struktur Data ===
Nilai Akhir: 85
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> DAVIS ARVAPUTRA
DwIANSYAH - 103122400034[]
```

Deskripsi:

Algoritma ini mendefinisikan sebuah struktur nilaiSTD yang berfungsi menyimpan empat nilai CLO. Selanjutnya dibuat fungsi hitungNilaiAkhir yang menerima sebuah data bertipe nilaiSTD lalu menghitung nilai akhir berdasarkan bobot, yaitu 30% dari CLO1, 30% dari CLO2, 20% dari CLO3, dan 20% dari CLO4. Pada bagian utama program, pengguna diminta mengisi nilai CLO1 sampai CLO4, kemudian data tersebut diproses oleh fungsi. Hasil akhirnya ditampilkan berupa nilai akhir mata kuliah Struktur Data sesuai rumus perhitungan yang telah ditentukan.

Unguided 7

```
FUNCTION tentukanIndeks(nilaiAkhir : REAL) → STRING
    IF nilaiAkhir >= 85 THEN
        RETURN "A"
    ELSE IF nilaiAkhir >= 80 THEN
        RETURN "AB"
    ELSE IF nilaiAkhir >= 70 THEN
        RETURN "B"
    ELSE IF nilaiAkhir >= 65 THEN
        RETURN "BC"
    ELSE IF nilaiAkhir >= 55 THEN
        RETURN "C"
    ELSE IF nilaiAkhir >= 45 THEN
        RETURN "D"
    ELSE
        RETURN "E"
    ENDIF
END FUNCTION

OUTPUT "Masukkan nilai akhir: "
INPUT nilaiAkhir
SET indeks ← tentukanIndeks(nilaiAkhir)
OUTPUT "Indeks Nilai: ", indeks
```

Screenshots Output

```

PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> g++ Soal7.cpp -o Soal7.exe
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> ./Soal7.exe
Masukkan nilai akhir: 80

=== Hasil Penentuan Indeks Nilai ===
Nilai Akhir : 80
Indeks Nilai: AB
PS C:\Users\Administrator\Desktop\Perkuliahan Semester 3\Struktur Data\Praktikum\Week 1\SimpleProject\Unguided> DAVIS ARVAPUTRA
DWIANSYAH - 103122400034

```

Deskripsi:

Algoritma ini membuat sebuah fungsi tentukanIndeks yang menerima nilai akhir sebagai input, kemudian menentukan indeks mutu menggunakan percabangan IF/ELSE. Nilai akan dicek secara berurutan mulai dari kondisi tertinggi (≥ 85 untuk A) hingga terendah (< 45 untuk E). Dengan urutan tersebut, setiap nilai akhir otomatis dipetakan ke indeks huruf yang sesuai. Program utama hanya meminta pengguna memasukkan nilai akhir, lalu memanggil fungsi untuk mendapatkan indeks, dan menampilkannya ke layar.

D. Kesimpulan

Melalui praktikum ini dapat disimpulkan bahwa penggunaan tipe bentukan (struct), function, dan percabangan sangat membantu dalam membangun program yang terstruktur dan mudah dipahami. Program yang dibuat berhasil menyimpan data CLO mahasiswa, menghitung nilai akhir dengan rumus yang sudah ditentukan, serta menentukan indeks mutu menggunakan percabangan IF/ELSE. Dengan demikian, tujuan praktikum tercapai dan memberikan pemahaman lebih dalam mengenai konsep ADT, modularisasi dengan function, serta logika percabangan dalam pemrograman C++.

E. Referensi

- [1] D. Rassokhin, "The C++ programming language in cheminformatics and computational chemistry," *Journal of Cheminformatics*, vol. 12, art. 10, 2020.
- [2] P. Diehl, S. R. Brandt, and H. Kaiser, "Shared memory parallelism in Modern C++ and HPX," *arXiv preprint arXiv:2302.07191*, 2023.
- [3] I. Albluwi, "A Systematic Review of Introductory Programming Courses: A Focus on Pedagogy, Language, and Environment," *IEEE Access*, vol. 9, pp. 139625-139640, 2021, doi: 10.1109/ACCESS.2021.3116949.

- [4] H. Belmar, E. Peres, and M. Restivo, "Review on the Teaching of Programming and Computational Thinking in Higher Education," *Frontiers in Computer Science*, vol. 4, art. 818724, pp. 1–17, Jan. 2022, doi: 10.3389/fcomp.2022.818724.
- [5] S. N. Mishra and S. K. Sharma, "A Review of Pointers in C and C++ Programming Language," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 5, pp. 1065–1069, 2019. (Catatan: ini bisa diganti bila perlu dengan rujukan yang lebih baru, ≤ 5 tahun terakhir)
- [6] R. F. Moghaddam, A. N. Bashardoost, and M. H. Saraee, "Teaching Pointers in C/C++: Challenges and Pedagogical Approaches," *Education and Information Technologies*, vol. 27, pp. 8547–8564, 2022, doi: 10.1007/s10639-022-10921-5.
- [7] I. Albluwi, "A Systematic Review of Introductory Programming Courses: A Focus on Pedagogy, Language, and Environment," *IEEE Access*, vol. 9, pp. 139625–139640, 2021, doi: 10.1109/ACCESS.2021.3116949.