**LAPORAN PRAKTIKUM**

**STRUKTUR DATA**

**MODUL I**

**PENGENALAN CODE BLOCKS**

****

# Disusun Oleh :

Abu Abdirrahman Humaid Al-Atsary  
103122400020

# Dosen

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

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO  
2025**

1. Dasar Teori

C++ merupakan bahasa pemrograman yang bersifat statically typed, multi-paradigm, dan general-purpose yang dikembangkan oleh Bjarne Stroustrup sebagai ekstensi dari bahasa C. Salah satu tujuan utama desain C++ adalah untuk menyediakan abstraksi tingkat tinggi dengan biaya performa yang minimal, menjadikannya bahasa yang sangat efisien dan memberikan kontrol tingkat rendah terhadap sumber daya sistem, terutama memori (Stroustrup, 2013). C++ mendukung berbagai paradigma pemrograman, termasuk pemrograman prosedural, berorientasi objek (OOP), dan generik, yang memberikan fleksibilitas bagi pengembang untuk memilih model yang paling sesuai untuk menyelesaikan masalah komputasi yang kompleks.

Paradigma yang paling menonjol dalam C++ adalah pemrograman berorientasi objek (OOP). Konsep ini memungkinkan pengembang untuk menstrukturkan program sebagai kumpulan objek yang saling berinteraksi. Tiga pilar utama dalam OOP yang diimplementasikan secara fundamental dalam C++ adalah encapsulation, inheritance, dan polymorphism. Encapsulation adalah mekanisme membungkus data (atribut) dan metode (fungsi) yang beroperasi pada data tersebut ke dalam satu unit tunggal yang disebut "kelas" atau class, sekaligus menyembunyikan detail implementasi internal dari dunia luar. Inheritance memungkinkan sebuah kelas baru (kelas turunan) untuk mewarisi properti dan perilaku dari kelas yang sudah ada (kelas dasar), yang mendorong penggunaan kembali kode (code reusability) dan penciptaan hierarki kelas. Sementara itu, polymorphism memungkinkan objek dari kelas yang berbeda untuk merespons pesan atau pemanggilan fungsi yang sama dengan cara yang spesifik untuk tipe mereka sendiri, sering kali diimplementasikan melalui virtual functions (Purwanto & Irawan, 2018).

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

Guided 1

|  |
| --- |
| *#include <iostream>*  *using namespace std;*  *int main() {*  *cout << "HELLO WORLD!" << endl;*  *return 0;*  *}* |

Screenshots Output



Deskripsi:

Pada kode "Hello, World!" adalah . Di C++, ini dicapai dengan menyertakan pustaka <iostream>, yang menyediakan std::cout untuk mengirim *output* ke konsol. Ini adalah langkah pertama dalam memahami konsep *stream I/O* (Input/Output) [3].

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 computer

AI-generated content may be incorrect.

Deskripsi:

Kode tersebut merupakan contoh penggunaan pointer di C++. Variabel x diisi dengann nilai 87, lalu alamat dari variabel x-nya disimpan di pointer px dengan px = &x. Kemudian nilai dari alamat yang ditunjuk px disalin ke y menggunakan y = \*px. Saat ditampilkan, &x menunjukkan alamat memori dari x, px menyimpan alamat x, \*px memberikan isi dari alamat tersebut (nilai x), dan y menyalin nilai itu.

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},*  *};*  *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

A computer screen shot of a black screen

AI-generated content may be incorrect.

Deskripsi :

1. Kode tersebut menggunakan array 1D di bagian nilai[MAX], yang dimana array tersebut digunakan untuk menyimpan input nilai siswa sebanyak 5 data, dan array 2D di bagian nilai\_tahun[MAX][MAX] yang telah diisi total 25. Pertama, program meminta user memasukkan 5 nilai lalu menampilkannya kembali. Setelah itu, program mencetak isi matriks nilai\_tahun baris per baris berdasarkan data yang sudah tersedia di array 2D tersebut.
2. 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  indeksNilai : string  END RECORD |

Deskripsi:

Kode tersebut merupakan sebuah deklarasi tipe Record bernama nilaiSTD yang berfunsi untuk mengelompokkan beberapa data terkait nilai mahasiswa dalam satu kesatuan. Pada kode tersebut juga ada 6 komponen, yaitu CLO1, CLO2, CLO3, CLO4, dan nilaiAkhir yang bertipe real, serta indexNilai yang bertipe string.

Unguided 2 :

|  |
| --- |
| VAR mhs1, mhs2 : nilaiSTD |

Deskripsi

Kode tersebut merupakan sebuah deklarasi 2 variable, yaitu mhs1 dan mhs2 yang masing-masing tipenya adalah nilaiSTD (tipe record).

Unguided 3

|  |
| --- |
| ALGORITHM inputNilai  VAR CLO1, CLO2, CLO3, CLO4 : real  BEGIN  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  END |

Deskripsi

Kode tersebut merupakan algoritma (inputNilai) yang berfunsi untuk meminta 4 inputan bilangan bertipe real.

Unguided 4

|  |
| --- |
| FUNCTION simpanNilai (CLO1, CLO2, CLO3, CLO4 : real) → nilaiSTD |

Deskripsi

Kode tersebut merupkan sebuah deklarasi sebuah fungsi untuk menerima parameter CLO1, CLO2, CLO3, dan CLO4. Kemudian, dari parameter tersebut disimpan dengan mengembalikan hasil berupa sebuah data bertipe nilaiSTD.

Unguided 5

|  |
| --- |
| FUNCTION simpanNilai (CLO1, CLO2, CLO3, CLO4 : real) → nilaiSTD  VAR data : nilaiSTD  BEGIN  data.CLO1 ← CLO1  data.CLO2 ← CLO2  data.CLO3 ← CLO3  data.CLO4 ← CLO4  data.nilaiAkhir ← 0  data.indeksNilai ← ""  RETURN data  END |

Deskripsi

Funsi simpanNilai di atas merupakan implementasi dari fungsi simpanNilai yang telah dideklarasi untuk menyimpan nilai CLO1, CLO2, CLO3, dan CLO4 ke dalam sebuah record nilaiSTD.

Unguided 6

|  |
| --- |
| FUNCTION hitungNilaiAkhir (CLO1, CLO2, CLO3, CLO4 : real) → real  VAR hasil : real  BEGIN  hasil ← (0.3 \* CLO1) + (0.3 \* CLO2) + (0.2 \* CLO3) + (0.2 \* CLO4)  RETURN hasil  END |

Deskripsi

Kode tersebut merupakan sebuah fungsi hitungNilaiAkhir yang digunakan untuk menghitung nilai akhir mata kuliah berdasarkan bobot dari empat nilai CLO. Fungsi tersebut melakukan operasi matematika, yaitu CLO1\*30% + CLO2\*30% + CLO3\*20% + CLO4\*20%, yang nantinya dimasukkan ke dalam variable hasil.

Unguided 7

|  |
| --- |
| FUNCTION tentukanIndeks (nilaiAkhir : real) → string  VAR indeks : string  BEGIN  IF nilaiAkhir >= 85 THEN  indeks ← "A"  ELSE IF nilaiAkhir >= 80 THEN  indeks ← "AB"  ELSE IF nilaiAkhir >= 70 THEN  indeks ← "B"  ELSE IF nilaiAkhir >= 65 THEN  indeks ← "BC"  ELSE IF nilaiAkhir >= 55 THEN  indeks ← "C"  ELSE IF nilaiAkhir >= 40 THEN  indeks ← "D"  ELSE  indeks ← "E"  ENDIF  RETURN indeks  END |

Deskripsi

Kode tersebut merupakan sebuah fungsi tentukanIndeks yang digunakan untuk mengubah nilai akhir menjadi nilai huruf berdasarkan rentang tertentu. Jika nilai akhir tinggi (≥85) maka hasilnya "A", sedangkan nilai lebih rendah dikategorikan bertahap menjadi "AB", "B", "BC", "C", "D", atau "E".

FULL KODE C++:

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  struct nilaiSTD {      float CLO1;      float CLO2;      float CLO3;      float CLO4;      float nilaiAkhir;      string indeksNilai;  };  nilaiSTD simpanNilai(float c1, float c2, float c3, float c4) {      nilaiSTD data;      data.CLO1 = c1;      data.CLO2 = c2;      data.CLO3 = c3;      data.CLO4 = c4;      data.nilaiAkhir = 0;      data.indeksNilai = "";      return data;  }  float hitungNilaiAkhir(float c1, float c2, float c3, float c4) {      return (0.3f \* c1) + (0.3f \* c2) + (0.2f \* c3) + (0.2f \* c4);  }  string tentukanIndeks(float nilaiAkhir) {      if (nilaiAkhir >= 85) return "A";      else if (nilaiAkhir >= 80) return "AB";      else if (nilaiAkhir >= 70) return "B";      else if (nilaiAkhir >= 65) return "BC";      else if (nilaiAkhir >= 55) return "C";      else if (nilaiAkhir >= 40) return "D";      else return "E";  }  int main() {      nilaiSTD mhs1, mhs2;      float c1, c2, c3, c4;      cout << "Masukkan nilai CLO-1 : ";      cin >> c1;      cout << "Masukkan nilai CLO-2 : ";      cin >> c2;      cout << "Masukkan nilai CLO-3 : ";      cin >> c3;      cout << "Masukkan nilai CLO-4 : ";      cin >> c4;      mhs1 = simpanNilai(c1, c2, c3, c4);      mhs1.nilaiAkhir = hitungNilaiAkhir(mhs1.CLO1, mhs1.CLO2, mhs1.CLO3, mhs1.CLO4);      mhs1.indeksNilai = tentukanIndeks(mhs1.nilaiAkhir);      cout << "\n=== HASIL PERHITUNGAN ===" << endl;      cout << "CLO1        : " << mhs1.CLO1 << endl;      cout << "CLO2        : " << mhs1.CLO2 << endl;      cout << "CLO3        : " << mhs1.CLO3 << endl;      cout << "CLO4        : " << mhs1.CLO4 << endl;      cout << "Nilai Akhir : " << mhs1.nilaiAkhir << endl;      cout << "Indeks Nilai: " << mhs1.indeksNilai << endl;      return 0;  } |

Screenshots Output

A screen shot of a computer

AI-generated content may be incorrect.

1. Kesimpulan
   1. Kesimpulan Guided

Program pertama memperkenalkan dasar I/O stream di C++ dengan mencetak “Hello, World!”, yang menjadi langkah awal memahami cara kerja output pada konsol. Program kedua menunjukkan konsep pointer, bagaimana sebuah variabel bisa diakses melalui alamat memorinya dengan operator & dan \*. Program ketiga memperlihatkan penggunaan array satu dimensi untuk menampung input nilai dari pengguna serta array dua dimensi untuk menyimpan dan menampilkan data statis berbentuk matriks. Secara keseluruhan, ketiga contoh program ini memberikan pemahaman bertahap mulai dari output sederhana, manipulasi memori dengan pointer, hingga pengolahan data menggunakan array 1D dan 2D, yang merupakan dasar penting dalam pemrograman C++.

* 1. Kesimpulan Unguided

Kesimpulannya, penerapan ADT (record/struct) di C++ memudahkan penyimpanan data mahasiswa secara terstruktur. Penggunaan fungsi simpanNilai, hitungNilaiAkhir, dan tentukanIndeks membuat program lebih modular, rapi, dan mudah dipahami. Program akhir dapat menerima input nilai, menghitung nilai akhir berdasarkan bobot, serta menampilkan indeks mutu mahasiswa sesuai rentang nilai.

1. Referensi[Bob

Purwanto, Y., & Irawan, B. (2018). Implementasi Konsep Object-Oriented Programming (OOP) untuk Sistem Informasi Geografis. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 5(2), 199-206.

Stroustrup, B. (2013). *The C++ Programming Language* (4th ed.). Addison-Wesley Professional.