

Ketentuan Penulisan Laporan adalah sebagai berikut

- a. Laporan Praktikum bersifat perorangan
- b. Waktu pengumpulan laporan adalah satu minggu atau H-1 Praktikum maksimal pukul pukul 23.59.
- c. Margin Halaman untuk Laporan Praktikum adalah Top: 4cm, Left: 4cm, Bottom: 3cm, Right:3cm.
- d. Laporan Praktikum mempunyai sistematika sebagai berikut:
 1. Halaman Judul/Cover
 - A. Dasar Teori
 - B. Guided (berisi screenshot source code & output program disertai penjelasannya file program .cpp dikumpulkan ke github sesuai jumlah guided tiap modul
 - C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya) file program .cpp dikumpulkan ke github sesuai jumlah unguided tiap
 - D. Kesimpulan
 - E. Referensi (jurnal 5 tahun terakhir)

*Screenshot wajib pakai watermark yang menerangkan NAMA dan NIM

- e. Laporan Praktikum yang tidak sesuai ketentuan dianggap tidak mengumpulkan.
- f. Plagiasi = 0.

Ketentuan Penilaian

- Telat 1 menit – 1jam = -5
- Telat 1 -7 hari = -20
- Telat 8- 14 hari = -40
- Telat 15-21 hari = -60
- Telat 22- 30 hari = -80
- Telat lebih dari 1 bulan = 0

g. Contoh Halaman Judul Laporan Praktikum

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Aditio Nugroho
NIM : 103122400008

Dosen

diah Septiana S.kom M.Cs

**PROGRAM STUDI
FAKULTAS
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

C++ adalah bahasa pemrograman multi-paradigma yang pertama kali dikembangkan oleh **Bjarne Stroustrup**. C++ menggabungkan fitur dari bahasa pemrograman prosedural (seperti C) dan paradigma berorientasi objek (OOP). Fitur kunci dari C++ meliputi enkapsulasi (encapsulation), pewarisan (inheritance), dan polimorfisme (polymorphism). Selain itu C++ mendukung generik programming (template), manipulasi memori tingkat rendah, dan kontrol yang lebih mendetail atas resource sistem, sehingga sering digunakan untuk aplikasi yang membutuhkan performa tinggi, seperti sistem tertanam, game, perangkat keras, dan aplikasi ilmiah.

Seiring perkembangan zaman, standar C++ diperbarui berkali-kali (C++11, C++14, C++17, C++20, dan seterusnya), menambahkan fitur-fitur modern seperti **Concepts**, **Ranges**, **Coroutines**, dan **Modules**. Perubahan ini menjadikan pengajaran dan penggunaan C++ tidak hanya dari sisi syntax dan struktur OOP tradisional, tetapi juga paradigma modern: generic programming, pemrograman paralel / kinerja tinggi, dan pemrograman lintas platform.

Untuk pengajaran C++, beberapa pendekatan yang disarankan adalah memperkenalkan terlebih dahulu dasar-dasar seperti tipe data, kontrol alur, fungsi dan kelas, lalu beranjak ke fitur-fitur modern, standar pustaka (STL), dan praktik terbaik (best practices) agar kode aman dan efisien.

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

Guided 1

```
1 main()
2   //Aditio Nugroho
3
4 #include <iostream>
5
6 using namespace std;
7
8 Qodo Gen: Options | Test this function
9 int main() {
10   cout << "Hello, World!" << endl;
11   return 0;
12 }
```

Screenshots Output

The screenshot shows a code editor interface with a dark theme. In the top-left, there's a file icon followed by the text "main()". Below it is a multi-line code editor containing the following C++ code:

```
1 //Aditio Nugroho
2
3 #include <iostream>
4
5 using namespace std;
6
7 Qodo Gen: Options | Test this function
8 int main() {
9     cout << "Hello, World!" << endl;
10    return 0;
11 }
```

Below the code editor, a terminal window is open with the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 5
Active code page: 65001
D:\KULIAH\semester 3>cd "d:\KULIAH\semester 3\Praktikum data\output"
d:\KULIAH\semester 3\Praktikum data\output>.\helloworld.exe
Hello, World!
```

Deskripsi:

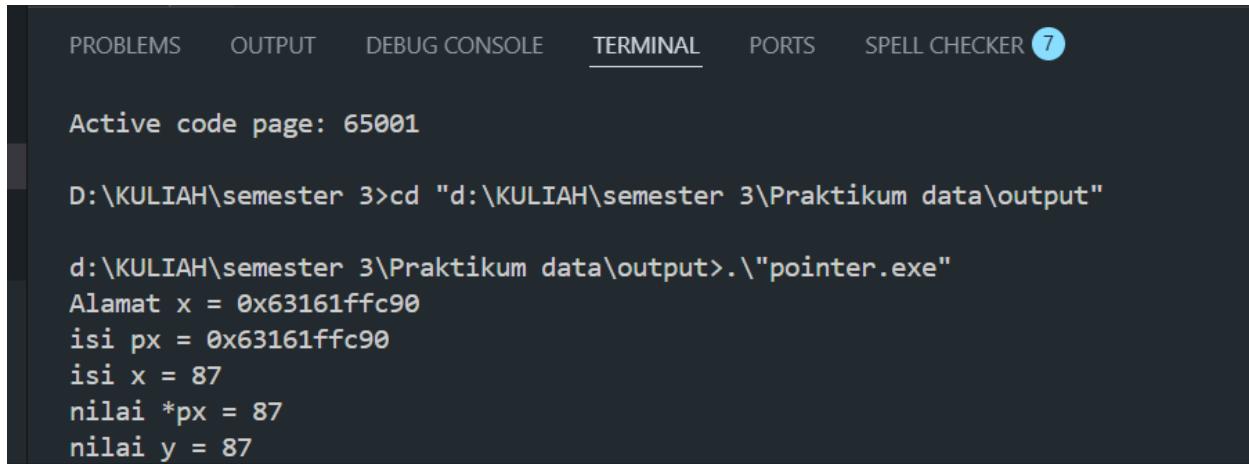
Program **Hello World** adalah program sederhana yang biasanya digunakan sebagai contoh pertama ketika belajar bahasa pemrograman. Tujuan utamanya adalah untuk memperkenalkan **struktur dasar program**, sintaksis, serta mekanisme input–output.

Guided 2

The screenshot shows a code editor interface with a dark theme. In the top-left, there's a file icon followed by the text "main()". Below it is a multi-line code editor containing the following C++ code:

```
1 //Aditio Nugroho_103122400008
2
3 #include <conio.h>
4 #include <iostream>
5
6 using namespace std;
7
8 Qodo Gen: Options | Test this function
9 int main() {
10     int x, y;
11     int *px;
12     x = 87;
13     px = &x;
14     y = *px;
15
16     cout << "Alamat x = " << &x << endl;
17     cout << "isi px = " << px << endl;
18     cout << "isi x = " << x << endl;
19     cout << "nilai *px = " << *px << endl;
20     cout << "nilai y = " << y << endl;
21
22     getch();
23     return 0;
24 }
```

Screenshots Output



```
Active code page: 65001
D:\KULIAH\semester 3>cd "d:\KULIAH\semester 3\Praktikum data\output"
d:\KULIAH\semester 3\Praktikum data\output>.\"pointer.exe"
Alamat x = 0x63161ffc90
isi px = 0x63161ffc90
isi x = 87
nilai *px = 87
nilai y = 87
```

Deskripsi:

Program ini merupakan contoh praktis yang menunjukkan hubungan erat antara array dan pointer di C++. Mengakses elemen array menggunakan pointer (*p) sering kali lebih efisien dan merupakan teknik yang umum digunakan dalam pemrograman *low-level* untuk memproses struktur data.

Guided 3

```
1 //Adition Nugroho_103122400008
2 #include <conio.h>
3 #include <iostream>
4 #define MAX 5
5
6 using namespace std;
7
8 Qodo Gen: Options | Test this function
9 int main(){
10     int i,j;
11     float nilai[MAX];
12     static int nilai_tahun[MAX][MAX] = {
13         {0,2,2,0,0},
14         {0,1,1,1,0},
15         {0,3,3,3,0},
16         {4,4,2,0,4},
17         {5,0,0,0,5},
18     };
19     for (i = 0; i<MAX; i++){
20         cout << "Masukan nilai ke-" <<i+1 << ": ";
21         cin >> nilai[i];
22     }
23
24     cout << "\nData nilai siswa:\n ";
25     for (i = 0; i<MAX;i++){
26         cout << "Nilai ke-" << i + 1 << "=" << nilai[i] << endl;
27     }
28
29     cout << "\nNilai tahunan:\n";
30     for(i = 0; i < MAX; i++){
31         for(j = 0; j < MAX; j++){
32             cout << nilai_tahun[i][j]<< " ";
33         }
34         cout << endl;
35     }
36
37     getch();
38     return 0;
39 }
```

Screenshots Output

```

d:\KULIAH\semester 3\Praktikum data\output> ."array.exe"
Masukan nilai ke-1: 1
Masukan nilai ke-2: 2
Masukan nilai ke-3: 3
Masukan nilai ke-4: 4
Masukan 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 ke-5=5

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

```

Deskripsi :

Program C++ ini menggunakan array satu dimensi dan array dua dimensi untuk menyimpan dan menampilkan data. Array satu dimensi (nilai[MAX]) digunakan untuk menampung nilai yang diinput oleh pengguna. Sementara itu, array dua dimensi (nilai_tahun[MAX][MAX]) berfungsi sebagai matriks untuk menyimpan data nilai tahunan yang sudah ditentukan sebelumnya.

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
  indeksNilai: string
END TYPE

```

Deskripsi:

Secara keseluruhan, definisi ini membuat sebuah **cetak biru** untuk sebuah objek atau entitas yang dapat menyimpan semua informasi terkait nilai mahasiswa dalam satu wadah yang

terorganisir. Ini mempermudah pengelolaan data yang saling berkaitan, seperti nilai CLO, nilai akhir, dan indeks nilai, tanpa harus menggunakan variabel yang terpisah.

Unguided 2

KAMUS

Mahasiswa1: nilaiSTD

Mahasiswa2: nilaiSTD

Deskripsi:

Kode tersebut mendeskripsikan sebuah kamus atau daftar yang berisi dua entri, yaitu Mahasiswa1 dan Mahasiswa2. Masing-masing entri ini memiliki tipe data nilaiSTD

Unguided 3

PROGRAM nilai_mahasiswa

KAMUS

clo1: real

clo2: real

clo3: real

clo4: real

ALGORITMA

PRINT "Masukan nilai CLO-1: "

READ clo1

PRINT "Masukan nilai CLO-1: "

READ clo12

PRINT "Masukan nilai CLO-1: "

READ clo12

PRINT "Masukan nilai CLO-1: "

READ clo12

END PROGRAM

D. Kesimpulan

Teks di atas adalah **pseudocode** dari sebuah program sederhana yang bertujuan untuk

menerima input nilai mahasiswa. Pseudocode ini menggunakan struktur dasar yang terdiri dari KAMUS (deklarasi variabel) dan ALGORITMA

Unguided 4

```
FUNCTION dataNILAISTD (clo1,clo2,clo3,clo4:real) nilaiSTD
```

Deskripsi:

Secara sederhana, fungsi ini bertanggung jawab untuk memproses keempat nilai input tersebut, melakukan perhitungan yang diperlukan, dan mengemas semua hasilnya ke dalam satu objek nilaiSTD yang kemudian akan dikembalikan ke bagian program yang memanggilnya. Fungsi ini membantu mengorganisasi kode dengan memisahkan logika pemrosesan data ke dalam blok yang terstruktur dan dapat digunakan kembali.

Unguided 5

```
FUNCTION dataNILAISTD (clo1, clo2, clo3, clo4: real) -> nilaiSTD  
DECLARE temporary_nilaiSTD:  
    nilaiSTD temporary_nilaiSTD.clo1 <- clo1  
    temporary_nilaiSTD.clo2 <- clo2  
    temporary_nilaiSTD.clo3 <- clo3  
    temporary_nilaiSTD.clo4 <- clo4  
    temporary_nilaiSTD.nilaiAkhir <- hitungNilaiAkhir(clo1, clo2, clo3, clo4)  
    temporary_nilaiSTD.indeksNilai <- hitungIndeksNilai(temporary_nilaiSTD.nilaiAkhir)  
RETURN temporary nilaiSTD  
END FUNCTION
```

Deskripsikan:

Fungsi dataNILAISTD adalah sebuah modul program yang dirancang untuk mengolah data nilai mahasiswa. Fungsi ini menerima empat parameter input yang mewakili nilai-nilai CLO, melakukan perhitungan nilai akhir dan indeks nilai, lalu mengembalikan semua data tersebut dalam satu kesatuan.

Unguided 6

```
FUNCTION hitungNilaiAkhir (clo1, clo2, clo3, clo4: real) -> real  
DECLARE  
    nilaiAkhir: real  
    nilaiAkhir <- (0.30 * clo1) + (0.30 * clo2) + (0.20 * clo3) + (0.20 * clo4)
```

```
RETURN nilaiAkhir  
END FUNCTION
```

Deskripsi:

Fungsi hitungNilaiAkhir merupakan modul yang dirancang secara spesifik untuk menghitung nilai rata-rata tertimbang dari empat komponen penilaian. Fungsi ini mengimplementasikan prinsip modularisasi dalam pemrograman, di mana sebuah tugas spesifik (perhitungan) dipisahkan ke dalam sebuah unit fungsional yang mandiri.

Unguided 7

```
FUNCTION hitungIndeksNilai (nilai: real) -> string  
DECLARE  
indeks: string IF (nilai > 80) THEN  
indeks <- "A"  
ELSE IF (nilai > 70 AND nilai <= 80) THEN  
indeks <- "AB" ELSE IF (nilai > 65 AND nilai <= 70) THEN  
indeks <- "B" ELSE IF (nilai > 60 AND nilai <= 65) THEN  
indeks <- "BC" ELSE IF (nilai > 50 AND nilai <= 60) THEN  
indeks <- "C" ELSE IF (nilai > 40 AND nilai <= 50) THEN  
indeks <- "D"  
ELSE  
indeks <- "E"  
END IF  
  
RETURN indeks  
END FUNCTION
```

Deskripsi:

Fungsi ini, bernama hitungIndeksNilai, adalah sebuah **kalkulator grade** otomatis. Fungsinya menerima satu nilai (nilai) dan, berdasarkan rentang nilai tersebut, ia akan mengembalikan sebuah huruf (indeks) seperti "A", "B", atau "E".

Logika di dalamnya menggunakan serangkaian **aturan jika-maka** (if-else if-else). Program akan memeriksa nilai secara berurutan, mulai dari yang tertinggi (di atas 80), sampai menemukan kondisi yang sesuai. Jika tidak ada kondisi yang cocok, ia akan mengembalikan nilai "E".

E. Referensi

- a. Cyganek, Bogusław. “*Modern C++ in the era of new technologies and challenges—why and how to teach modern C++?*” Proceedings of the 17th Conference on Computer Science and Intelligence Systems, Vol. 30, pages 35–40, 2022.
- b. Thoman, P., et al. “*The Celerity High-level API: C++20 for Accelerator Clusters*” (Springer, 2022).
- c. Hollman, D. S., Adelstein Lelbach, B., Edwards, H. C., et al. “*mspan in C++: A Case Study in the Integration of Performance Portable Features into International Language Standards*”, 2020.