

**LAPORAN PRAKTIKUM MODUL 2**  
**PENGENALAN C++ (BAGIAN 2)**



**Disusun Oleh:**  
**Ade Fatkhul Anam 2211104051**

**Dosen :**  
**Wahyu Andi Saputra, S.Pd., M.Eng.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**PURWOKERTO**  
**2024**

## MODUL 2

### PENGENALAN BAHASA C++ (BAGIAN KEDUA)

#### 1. Tujuan

- Memahami penggunaan pointer dan alamat memori dalam program.
- Mengimplementasikan fungsi dan prosedur pada bahasa pemrograman C++.
- Menggunakan array satu dimensi, dua dimensi, dan berdimensi banyak untuk menyimpan dan mengakses data.
- Mempelajari hubungan antara pointer dan array dalam konteks pengelolaan data.
- Mempelajari cara kerja string dalam bahasa C++ serta kaitannya dengan pointer.
- Mempelajari cara melewatkan parameter pada fungsi dengan metode **Call by Value**, **Call by Pointer**, dan **Call by Reference**

#### 2. Landasan Teori

- Bahasa C++
- Array

Array adalah kumpulan elemen bertipe data yang sama. Terdiri dari:

  - Array Satu Dimensi: Digunakan untuk menyimpan data dalam satu baris.
  - Array Dua Dimensi: Berbentuk tabel dengan baris dan kolom.
  - Array Berdimensi Banyak: Array dengan lebih dari dua dimensi untuk penyimpanan data yang lebih kompleks.
- Pointer

Pointer adalah variabel yang menyimpan alamat memori variabel lain. Digunakan untuk manipulasi memori secara efisien. Operasi pointer memungkinkan akses langsung ke nilai variabel yang ditunjuk.
- Hubungan Pointer dan Array

Array bertindak seperti pointer ke elemen pertama. Operasi pada array dapat dilakukan melalui pointer.
- String

String adalah array karakter yang diakhiri dengan `\0`. String dapat diproses seperti array dan sering digunakan untuk memanipulasi data teks.
- Fungsi

Fungsi adalah blok kode yang melaksanakan tugas tertentu dan dapat menerima parameter serta mengembalikan nilai. Fungsi mempermudah pengorganisasian program.
- Prosedur

Prosedur adalah fungsi yang tidak mengembalikan nilai (menggunakan `void`). Digunakan untuk melakukan tugas tanpa hasil balik.
- Parameter Fungsi

Call by Value: Mengirim salinan nilai.  
Call by Pointer: Mengirim alamat memori, memungkinkan perubahan langsung.  
Call by Reference: Mengirim referensi, mirip dengan pointer tetapi lebih sederhana dalam sintaks.

### 3. Guided

#### a) Array

Array merupakan kumpulan data dengan nama yang sama dan setiap elemen bertipe data sama. Untuk mengakses setiap komponen / elemen array berdasarkan indeks dari setiap elemen.

1. Array satu Dimensi

```
int nilai[10] = {1, 2, 3, 4, 5}; // Array 1D
```

2. Array Dua Dimensi

```
int data_nilai[4][3]; // Array 2D
```

3. Array Berdimensi Banyak

```
int data_rumit[4][6][6]; // Array banyak dimensi
```

Output:

```
d:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\modul2.cpp:23:9: warning: unused variable 'nilai' [-Wunused-variable]
 23 |     int nilai[10] = {1, 2, 3, 4, 5}; // Array 1D
    |     ~~~~~
d:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\modul2.cpp:24:9: warning: unused variable 'data_nilai' [-Wunused-variable]
 24 |     int data_nilai[4][3]; // Array 2D
    |     ~~~~~
d:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\modul2.cpp:25:9: warning: unused variable 'data_rumit' [-Wunused-variable]
 25 |     int data_rumit[4][6][6]; // Array banyak dimensi
```

#### b) Pointer

Pointer adalah variabel yang menyimpan alamat memori variabel lain. Digunakan untuk manipulasi memori secara efisien. Operasi pointer memungkinkan akses langsung ke nilai variabel yang ditunjuk.

```
// POINTER
```

```
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 yang ditunjuk px = " << *px << endl;
```

```
cout << "Nilai y = " << y << endl;
```

Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> cd 'd:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output'
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> & .\modul2.exe
Alamat x = 0xe52b5ffdd0
Isi px = 0xe52b5ffdd0
Isi x = 87
Nilai yang ditunjuk px = 87
Nilai y = 87
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output>

```

**c) Fungsi**

Fungsi adalah blok kode yang melaksanakan tugas tertentu dan dapat menerima parameter serta mengembalikan nilai. Fungsi mempermudah pengorganisasian program.

```
// Fungsi
```

```
int penjumlahan(int a, int b){
    return a + b;
}

void greet(string name) {
    cout << "Hello, " << name << "!" << endl;
}

int main() {
    int hasil = penjumlahan(5, 3);

    cout << "hasil " << hasil << endl;

    greet("alice");
}
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> cd 'd:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output'
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> & .\modul2.exe
hasil 8
Hello, alice!
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> 
```

**d) Prosedur**

```
// Prosedur
#include <iostream>
#include <stdlib.h> // untuk system("cls")

using namespace std;

/* prototype fungsi */
void tulis(int x);

int main() {
    system("cls"); // menggunakan huruf kecil
    int jum;
    cout << "Jumlah baris kata: ";
    cin >> jum;
    tulis(jum);
    cin.get(); // menunggu input agar konsol tidak langsung
    tertutup
    cin.get(); // digunakan dua kali karena cin.get() menangkap
    enter sebelumnya
    return 0;
}

/* badan prosedur */
void tulis(int x) {
    for (int i = 0; i < x; i++) {
        cout << "Baris ke-" << i + 1 << endl;
    }
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Jumlah baris kata: 10
Baris ke-1
Baris ke-2
Baris ke-3
Baris ke-3
Baris ke-3
Baris ke-4
Baris ke-5
Baris ke-6
Baris ke-7
Baris ke-8
Baris ke-9
Baris ke-10
x
```

#### 4. UNGUIDED

1. Buatlah program untuk menampilkan Output seperti berikut dengan data yang diinputkan oleh user!

Code:

```
#include <iostream>
#include <stdlib.h>
using namespace std;

int main() {
    const int n = 10; // Panjang array
    int dataArray[n];

    // Meminta input dari pengguna untuk mengisi array
    cout << "Masukkan " << n << " angka: ";
    for (int i = 0; i < n; i++) {
        cin >> dataArray[i];
    }

    // Menampilkan seluruh elemen array
    cout << "Data Array : ";
    for (int i = 0; i < n; i++) {
        cout << dataArray[i] << " ";
    }
    cout << endl;

    // Menampilkan elemen genap
    cout << "Nomor Genap : ";
    for (int i = 0; i < n; i++) {
        if (dataArray[i] % 2 == 0) {
            cout << dataArray[i] << ", ";
        }
    }
    cout << endl;

    // Menampilkan elemen ganjil
    cout << "Nomor Ganjil : ";
    for (int i = 0; i < n; i++) {
        if (dataArray[i] % 2 != 0) {
            cout << dataArray[i] << ", ";
        }
    }
    cout << endl;

    return 0;
}
```

## Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> cd 'd:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output'
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> & .\modul2.exe
Masukkan 10 angka: 11 12 13 14 15 16 17 18 19 20
Data Array : 11 12 13 14 15 16 17 18 19 20
Nomor Genap : 12, 14, 16, 18, 20,
Nomor Ganjil : 11, 13, 15, 17, 19,
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> █
```

2. Buatlah program Input array tiga dimensi tetapi jumlah atau ukuran elemennya diinputkan oleh user!

Code:

```
#include <iostream>
#include <stdlib.h>
using namespace std;

int main() {
    int x, y, z;

    // Meminta ukuran dimensi array dari pengguna
    cout << "Masukkan ukuran dimensi pertama: ";
    cin >> x;
    cout << "Masukkan ukuran dimensi kedua: ";
    cin >> y;
    cout << "Masukkan ukuran dimensi ketiga: ";
    cin >> z;

    // Deklarasi array tiga dimensi dengan ukuran yang ditentukan
    oleh pengguna
    int array3D[x][y][z];

    // Input elemen-elemen array
    cout << "Masukkan elemen array:\n";
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                cout << "Elemen [" << i << "][" << j << "][" << k <<
                "]: ";
                cin >> array3D[i][j][k];
            }
        }
    }

    // Menampilkan elemen-elemen array
    cout << "\nElemen-elemen array tiga dimensi:\n";
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                cout << "Elemen [" << i << "][" << j << "][" << k <<
                "] = " << array3D[i][j][k] << endl;
            }
        }
    }
}
```

```

    }
}

return 0;
}

```

Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> cd 'd:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output'
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> & .\'modul2.exe'
Masukkan ukuran dimensi pertama: 4
Masukkan ukuran dimensi kedua: 2
Masukkan ukuran dimensi ketiga: 1
Masukkan elemen array:
Elemen [0][0][0]: 1 2 3
Elemen [0][1][0]: Elemen [1][0][0]: Elemen [1][1][0]: 3 2 1
Elemen [2][0][0]: Elemen [2][1][0]: Elemen [3][0][0]: 213
Elemen [3][1][0]: 321

Elemen-elemen array tiga dimensi:
Elemen [0][0][0] = 1
Elemen [0][1][0] = 2
Elemen [1][0][0] = 3
Elemen [1][1][0] = 3
Elemen [2][0][0] = 2
Elemen [2][1][0] = 1
Elemen [3][0][0] = 213
Elemen [3][1][0] = 321
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output>

```

3. Buatlah program menu untuk mencari nilai Maksimum, Minimum dan Nilai rata – rata dari suatu array dengan input yang dimasukan oleh user!

Code:

```

#include <iostream>
#include <stdlib.h>
using namespace std;

// Fungsi untuk mencari nilai maksimum
int cariMaksimum(int array[], int n) {
    int maks = array[0];
    for (int i = 1; i < n; i++) {
        if (array[i] > maks) {
            maks = array[i];
        }
    }
    return maks;
}

// Fungsi untuk mencari nilai minimum
int cariMinimum(int array[], int n) {
    int min = array[0];
    for (int i = 1; i < n; i++) {
        if (array[i] < min) {
            min = array[i];
        }
    }
}

```



```

    }
    return min;
}

// Fungsi untuk mencari nilai rata-rata
float cariRataRata(int array[], int n) {
    int total = 0;
    for (int i = 0; i < n; i++) {
        total += array[i];
    }
    return (float)total / n;
}

int main() {
    int n;

    // Meminta jumlah elemen array
    cout << "Masukkan jumlah elemen array: ";
    cin >> n;

    int array[n];

    // Meminta input dari user untuk mengisi array
    cout << "Masukkan elemen-elemen array: ";
    for (int i = 0; i < n; i++) {
        cin >> array[i];
    }

    int pilihan;
    do {
        // Menampilkan menu
        cout << "\nMenu:\n";
        cout << "1. Cari Nilai Maksimum\n";
        cout << "2. Cari Nilai Minimum\n";
        cout << "3. Cari Nilai Rata-rata\n";
        cout << "4. Keluar\n";
        cout << "Pilih menu (1-4): ";
        cin >> pilihan;

        // Menjalankan pilihan user
        switch (pilihan) {
            case 1:
                cout << "Nilai maksimum: " << cariMaksimum(array, n) <<
endl;
                break;
            case 2:
                cout << "Nilai minimum: " << cariMinimum(array, n) <<
endl;
                break;
            case 3:

```

```

        cout << "Nilai rata-rata: " << cariRataRata(array, n) <<
endl;
        break;
    case 4:
        cout << "Keluar dari program." << endl;
        break;
    default:
        cout << "Pilihan tidak valid. Silakan coba lagi." <<
endl;
    }

    } while (pilihan != 4);

    return 0;
}

```

### Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> cd 'd:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output'
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output> & .\modul2.exe
Masukkan jumlah elemen array: 5
Masukkan elemen-elemen array: 3
123
12345
123
12334

Menu:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar
Pilih menu (1-4): 1
Nilai maksimum: 12345

Menu:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar
Pilih menu (1-4): 2
Nilai minimum: 3

Menu:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar
Pilih menu (1-4): 3
Nilai rata-rata: 4985.6

Menu:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar
Pilih menu (1-4): 4
Keluar dari program.
PS D:\PRAKTIKUM DATA STRUCTURE\PRAKTIKUM\pertemuan3\output>

```

### Kesimpulan:

Praktikum kali ini memungkinkan pengguna untuk melakukan analisis sederhana pada array dengan mencari nilai maksimum, minimum, dan rata-rata dari elemen yang diinputkan. Pengguna diminta untuk memasukkan jumlah elemen dan nilai-nilainya, setelah itu program menampilkan menu untuk memilih operasi yang diinginkan. Dengan memanfaatkan fungsi terpisah untuk setiap operasi, program ini memastikan struktur yang jelas dan mudah dipahami. Kompleksitas waktu  $O(n)$  untuk setiap fungsi menjamin efisiensi dalam pengolahan data. Secara keseluruhan, program ini adalah alat yang berguna untuk analisis data numerik secara intuitif dan interaktif.