

# **LAPORAN PRAKTIKUM**

## **Modul 1**

### **Pengenalan CPP bagian 1**



**Disusun Oleh:**

**Yogi Hafidh Maulana - 2211104061**

**SE06-02**

**Dosen :**

**Wahyu Andi**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY**

**PURWOKERTO**

**2024**

## 1. Tujuan

- Memahami konsep dasar dan penerapan array satu dimensi, dua dimensi, dan multi dimensi dalam C++.
- Menguasai penggunaan pointer untuk memanipulasi alamat memori variabel.
- Mengimplementasikan fungsi dan prosedur dengan parameter call by value, call by pointer, dan call by reference.
- Mengelola memori secara efisien menggunakan pointer dan array.
- Mengaplikasikan konsep array dan pointer dalam membangun program yang terstruktur dan modular.

## 2. Landasan Teori

- Array

Array adalah struktur data yang menyimpan kumpulan elemen bertipe data sama dalam satu variabel dengan indeks tertentu. Array digunakan untuk mengelola sejumlah besar data secara terstruktur dan efisien. Terdapat beberapa jenis array seperti array satu dimensi, dua dimensi, dan multi dimensi. Array satu dimensi menyimpan data dalam bentuk linear, sementara array dua dimensi atau lebih dapat digunakan untuk merepresentasikan data dalam bentuk tabel atau matriks.

- Pointer

Pointer merupakan variabel khusus yang menyimpan alamat memori dari variabel lain. Pointer memungkinkan akses langsung ke alamat memori, sehingga perubahan pada variabel yang ditunjuk oleh pointer akan berdampak pada variabel aslinya. Pointer dapat digunakan untuk memanipulasi array dan variabel lain secara lebih efisien. Dalam C++, pointer juga memiliki hubungan erat dengan array, di mana nama array sendiri merupakan pointer ke elemen pertama array.

- Fungsi dan Prosedur

Fungsi dan prosedur dalam C++ adalah blok kode yang dirancang untuk melakukan tugas tertentu, membantu memecah program menjadi bagian-bagian yang lebih kecil dan terstruktur. Fungsi digunakan untuk mengembalikan nilai, sedangkan prosedur (dalam bentuk fungsi void) tidak mengembalikan nilai. Fungsi dapat menerima parameter melalui call by value, call by pointer, atau call by reference. Call by value hanya melewatkan nilai, sehingga tidak mempengaruhi variabel asli. Call by pointer dan call by reference memungkinkan fungsi untuk memanipulasi variabel asli karena keduanya melewatkan alamat memori variabel tersebut.

### 3. Guided

#### a) Array 1 Dimensi

Array 1 Dimensi adalah array yang hanya terdiri dari satu larik data saja. Dalam C++ data array disimpan dalam memori pada lokasi yang berurutan. Elemen pertama memiliki indeks 0 dan elemen selanjutnya memiliki indeks 1 dan seterusnya. Jadi jika terdapat array dengan 5 elemen maka elemen pertama memiliki indeks 0 dan elemen terakhir memiliki indeks 4.

```
23      ....// ARRAY 1D
24      ....int nilai[10] = {1, 2, 3, 4, 5};
25
26      ....// Pemanggilan Array menggunakan index
27      ....cout << nilai[0];
28      ....cout << nilai[1];
29      ....cout << nilai[2];
30      ....cout << nilai[3];
31      ....cout << nilai[4];
32
33      ....// Pemanggilan arraya menggunakan Looping
34      ....for(int i = 0; i < 5; i++) {
35      ....    ....cout << nilai[i] << endl;
36      ....}
```

12345

PS D:\PROJECT\C++ Project\Laprak 2>

1

2

3

4

5

PS D:\PROJECT\C++ Project\Laprak 2>

## b) Array 2D dan Banyak Dimensi

Array dua dimensi mirip seperti tabel. Jadi array dua dimensi bisa digunakan untuk menyimpan data dalam bentuk tabel. Terbagi menjadi dua bagian, dimensi pertama dan dimensi kedua. Cara akses, deklarasi, inisialisasi, dan menampilkan data sama dengan array satu dimensi, hanya saja indeks yang digunakan ada dua.

Array banyak dimensi merupakan array yang mempunyai indeks banyak, lebih dari dua. Indeks inilah yang menyatakan dimensi array. Array berdimensi banyak lebih susah dibayangkan, sejalan dengan jumlah dimensi dalam array

```
38 // ARRAY 2D
39 int data_nilai[3][4] = {{1,2,3,4}, {4, 6, 7, 8}, {9, 10, 11, 12}};
40
41 for (int i = 0; i < 3; i++){
42     for (int j = 0; j < 4; j++){
43         cout << data_nilai[i][j] << " ";
44     }
45     cout << endl;
46 }
47
48 // ARRAY BANYAK DIMENSI
49 int data_rumit[4][6][6];
```

```
1 2 3 4
4 6 7 8
9 10 11 12
PS D:\PROJECT\C++ Project\Laprak 2>
```

### c) Pointer

Semua data yang ada digunakan oleh program komputer disimpan di dalam memori (RAM) komputer. Memori dapat digambarkan sebagai sebuah array 1 dimensi yang berukuran sangat besar. Seperti layaknya array, setiap cell memory memiliki “indeks” atau “alamat” unik yang berguna untuk identitas yang biasa kita sebut sebagai “address” Saat program berjalan, Sistem Operasi (OS) akan mengalokasikan space memory untuk setiap variabel, objek, atau array yang kita buat.

Variabel pointer merupakan dasar tipe variabel yang berisi integer dalam format heksadesimal. Pointer digunakan untuk menyimpan alamat memori variabel lain sehingga pointer dapat mengakses nilai dari variabel yang alamatnya ditunjuk.

```
51  ....// POINTER
52  ....int x, y;
53  ....int *px;
54  ....x= 87;
55  ....px = &x;
56  ....y= *px;
57
58  ....cout << "Alamat x: " << &x << endl;
59  ....cout << "Isi px: " << px << endl;
60  ....cout << "Isi X: " << x << endl;
61  ....cout << "Nilai yang ditunjuk px: " << *px << endl;
62  ....cout << "Nilai y: " << y << endl;
63  ....getch();
64
65  ....return 0;
```

```
Alamat x: 0x5ffe90
Isi px: 0x5ffe90
Isi X: 87
Nilai yang ditunjuk px: 87
Nilai y: 87
```

#### d) Fungsi dan Prosedur

Fungsi merupakan blok dari kode yang dirancang untuk melaksanakan tugas khusus dengan tujuan, pertama program menjadi terstruktur, sehingga mudah dipahami dan mudah dikembangkan. Program dibagi menjadi beberapa modul yang kecil. Kedua dapat mengurangi pengulangan kode (duplikasi kode) sehingga menghemat ukuran program. Pada umumnya fungsi memerlukan masukan yang dinamakan sebagai parameter. Masukan ini selanjutnya diolah oleh fungsi. Hasil akhir fungsi berupa sebuah nilai (nilai balik fungsi).

Dalam C sebenarnya tidak ada prosedur, semua berupa fungsi, termasuk main() pun adalah sebuah fungsi. Jadi prosedur dalam C merupakan fungsi yang tidak mengembalikan nilai, biasa diawali dengan kata kunci void di depan nama prosedur.

```
5 // FUNGSI
   Tabnine | Edit | Test | Explain | Document | Ask
6 int penjumlahan(int a, int b){
7     ... return a + b;
8 }
9
10 // PROSEDUR
   Tabnine | Edit | Test | Explain | Document | Ask
11 void greet(string name){
12     ... cout << "Hallo, " << name << "!" << endl;
13 }
14
```

```
Hasil: 8
Hallo, John Doe!
PS D:\PROJECT\C++ Project\Laparak 2>
```

#### 4. Unguided

- a) Buatlah program untuk menampilkan Output seperti berikut dengan data yang diinputkan oleh user!

```
Data Array : 1 2 3 4 5 6 7 8 9 10
Nomor Genap : 2, 4, 6, 8, 10,
Nomor Ganjil : 1, 3, 5, 7, 9,
```

Code

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> dataArray; // Untuk menyimpan data array
    vector<int> nomorGenap; // Untuk menyimpan bilangan genap
    vector<int> nomorGanjil; // Untuk menyimpan bilangan ganjil
    int n;

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

    // Mengisi array dengan angka berurutan mulai dari 0
    for (int i = 0; i < n; i++) {
        dataArray.push_back(i); // Menambahkan angka ke dalam dataArray
        // Memisahkan angka genap dan ganjil
        if (i % 2 == 0) {
            nomorGenap.push_back(i);
        } else {
            nomorGanjil.push_back(i);
        }
    }

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

    // Menampilkan nomor genap
    cout << "Nomor Genap : ";
    for (int i = 0; i < nomorGenap.size(); i++) {
        cout << nomorGenap[i];
        if (i != nomorGenap.size() - 1) {
            cout << ", ";
        }
    }
    cout << ", " << endl;

    // Menampilkan nomor ganjil
    cout << "Nomor Ganjil : ";
    for (int i = 0; i < nomorGanjil.size(); i++) {
        cout << nomorGanjil[i];
        if (i != nomorGanjil.size() - 1) {
            cout << ", ";
        }
    }
    cout << ", " << endl;

    return 0;
}
```

## Output

```
Masukkan jumlah elemen data array: 10
Masukkan 10 elemen: 1 2 3 4 5 6 7 8 9 10
Data Array : 1 2 3 4 5 6 7 8 9 10
Nomor Genap : 2, 4, 6, 8, 10,
Nomor Ganjil : 1, 3, 5, 7, 9,
PS D:\PROJECT\C++ Project\Laprak 2>
```

- b) Buatlah program Input array tiga dimensi tetapi jumlah atau ukuran elemennya diinputkan oleh user!

Code:

```
#include <iostream>
using namespace std;

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

    // Meminta input ukuran dimensi array
    cout << "Masukkan ukuran dimensi pertama (x): ";
    cin >> x;
    cout << "Masukkan ukuran dimensi kedua (y): ";
    cin >> y;
    cout << "Masukkan ukuran dimensi ketiga (z): ";
    cin >> z;

    // Mendeklarasikan array tiga dimensi
    int array3D[x][y][z];

    // Mengisi array tiga dimensi dengan nilai urut atau input dari user
    cout << "Masukkan nilai untuk setiap 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 << "Nilai untuk array[" << i << "][" << j << "][" << k <<
                "]" : ";
                cin >> array3D[i][j][k];
            }
        }
    }

    // Menampilkan isi array tiga dimensi
    cout << "\nArray 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 << "array[" << i << "][" << j << "][" << k << "] = " <<
                array3D[i][j][k] << endl;
            }
        }
    }

    return 0;
}
```



Output:

```
Masukkan ukuran dimensi pertama (x): 2
Masukkan ukuran dimensi kedua (y): 2
Masukkan ukuran dimensi ketiga (z): 2
Masukkan nilai untuk setiap elemen array:
Nilai untuk array[0][0][0] : 1 2 3
Nilai untuk array[0][0][1] : Nilai untuk array[0][1][0] : Nilai untuk array[0][1][1] : 4 5 6
Nilai untuk array[1][0][0] : Nilai untuk array[1][0][1] : Nilai untuk array[1][1][0] : 7 8 9
Nilai untuk array[1][1][1] :
Array Tiga Dimensi:
array[0][0][0] = 1
array[0][0][1] = 2
array[0][1][0] = 3
array[0][1][1] = 4
array[1][0][0] = 5
array[1][0][1] = 6
array[1][1][0] = 7
array[1][1][1] = 8
PS D:\PROJECT\C++ Project\Labrak 2>
```

- c) 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 <vector>
using namespace std;

// Fungsi untuk mencari nilai maksimum
int cariMaksimum(vector<int> &arr) {
    int maksimum = arr[0];
    for (int i = 1; i < arr.size(); i++) {
        if (arr[i] > maksimum) {
            maksimum = arr[i];
        }
    }
    return maksimum;
}

// Fungsi untuk mencari nilai minimum
int cariMinimum(vector<int> &arr) {
    int minimum = arr[0];
    for (int i = 1; i < arr.size(); i++) {
        if (arr[i] < minimum) {
            minimum = arr[i];
        }
    }
    return minimum;
}

// Fungsi untuk mencari nilai rata-rata
double cariRataRata(vector<int> &arr) {
    int jumlah = 0;
    for (int i = 0; i < arr.size(); i++) {
        jumlah += arr[i];
    }
    return static_cast<double>(jumlah) / arr.size();
}

int main() {
    vector<int> dataArray;
    int n, pilihan, nilai;

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

    // Memasukkan nilai elemen array
    cout << "Masukkan " << n << " elemen array:\n";
    for (int i = 0; i < n; i++) {
        cin >> nilai;
        dataArray.push_back(nilai);
    }

    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 opsi (1-4): ";
        cin >> pilihan;

        // Menentukan aksi berdasarkan pilihan menu
        switch (pilihan) {
            case 1:
                cout << "Nilai Maksimum: " << cariMaksimum(dataArray) << endl;
                break;
            case 2:
                cout << "Nilai Minimum: " << cariMinimum(dataArray) << endl;
                break;
            case 3:
                cout << "Nilai Rata-rata: " << cariRataRata(dataArray) << endl;
                break;
            case 4:
                cout << "Keluar dari program.\n";
                break;
            default:
                cout << "Pilihan tidak valid! Silakan pilih lagi.\n";
                break;
        }
    } while (pilihan != 4);

    return 0;
}
```

Output:

```
Masukkan jumlah elemen array: 10
Masukkan 10 elemen array:
9 10 45 78 23 2 99 25 75 30
```

```
MENU
```

1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar

```
Pilih opsi (1-4): 1
```

```
Nilai Maksimum: 99
```

```
MENU
```

1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar

```
Pilih opsi (1-4): 2
```

```
Nilai Minimum: 2
```

```
MENU
```

1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar

```
Pilih opsi (1-4): 3
```

```
Nilai Rata-rata: 39.6
```

```
MENU
```

1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar

```
Pilih opsi (1-4): 4
```

```
Keluar dari program.
```

```
PS D:\PROJECT\C++ Project\Laprak 2> █
```

## 5. Kesimpulan

Modul 2 ini membahas konsep dasar struktur data dalam bahasa C++, termasuk penggunaan array, pointer, dan implementasi fungsi serta prosedur untuk memanipulasi data secara efisien. Array digunakan untuk menyimpan kumpulan data bertipe sama, sementara pointer berfungsi menyimpan alamat memori dan memungkinkan akses serta modifikasi langsung terhadap data. Modul ini juga menjelaskan hubungan antara array dan pointer, serta metode melewatkan parameter ke fungsi melalui `call by value`, `call by pointer`, dan `call by reference` yang memungkinkan pengelolaan memori secara fleksibel. Dengan memahami teknik ini, pemrogram dapat menyusun program yang lebih terstruktur, efisien, dan mudah dikelola untuk memecahkan berbagai masalah komputasi.

