

# **LAPORAN PRAKTIKUM**

## **Modul 2**

### **Pengenalan CPP bagian 2**



**Disusun Oleh:**

**Satria Ariq Adelard Dompas - 2211104033**

**SE06-02**

**Dosen : Wahyu Andi Saputra**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY**

**PURWOKERTO**

**2024**

## 1. Tujuan

- Memahami konsep dasar dan implementasi array satu dimensi, dua dimensi, dan multidimensi pada C++.
- Kuasai cara menggunakan pointer untuk memanipulasi alamat memori variabel.
- Mengimplementasikan fungsi dan prosedur dengan parameter call-by-value, call-by-pointer, dan call-by-reference.
- Manajemen memori yang efisien menggunakan pointer dan array.
- Menerapkan konsep array dan pointer dalam membangun program terstruktur dan modular.

## 2. Landasan Teori

- Array

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

- Pointer

Pointer merupakan variabel khusus yang menyimpan alamat memori variabel lain. Pointer memungkinkan akses langsung ke alamat memori, sehingga perubahan pada variabel yang ditunjuk oleh pointer akan mempengaruhi variabel aslinya. Pointer dapat digunakan untuk memanipulasi array dan variabel lain dengan lebih efisien. Dalam C++, pointer juga mempunyai hubungan erat dengan array, dimana nama array itu 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 membagi program menjadi bagian-bagian yang lebih kecil dan terstruktur. Fungsi digunakan untuk mengembalikan nilai, sedangkan prosedur (dalam bentuk fungsi kosong) tidak mengembalikan nilai. Fungsi dapat menerima parameter melalui panggilan demi nilai, panggilan demi penunjuk, atau panggilan demi referensi. Memanggil berdasarkan nilai hanya meneruskan nilai sehingga tidak mempengaruhi variabel asli. Panggilan dengan penunjuk dan panggilan dengan referensi memungkinkan fungsi memanipulasi variabel nyata karena meneruskan alamat memori variabel.

### 3. Guided

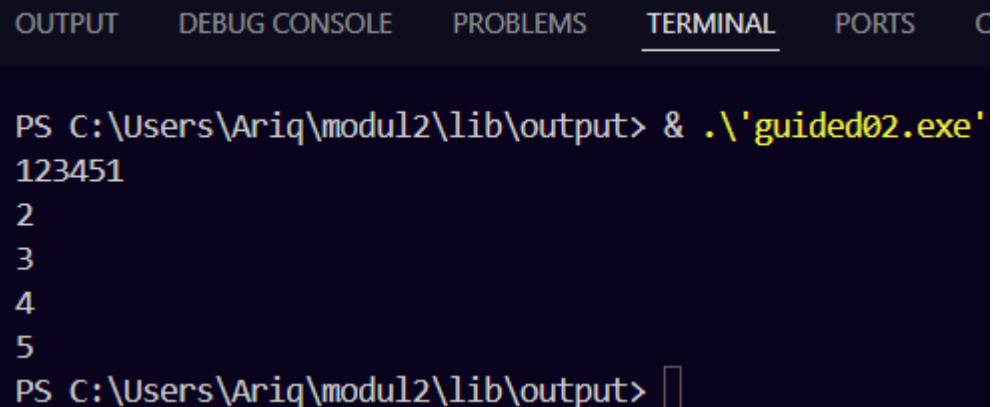
#### a) Array 1 Dimensi

Tabel 1 dimensi adalah tabel yang terdiri dari array data. Di C++, data array disimpan dalam memori di lokasi berurutan. Elemen pertama memiliki indeks 0 dan elemen berikutnya memiliki indeks 1, dan seterusnya. Jadi jika ada array dengan 5 elemen maka elemen pertama memiliki indeks 0 dan elemen terakhir memiliki indeks 4.

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

// Pemanggilan Array menggunakan index
cout << nilai[0];
cout << nilai[1];
cout << nilai[2];
cout << nilai[3];
cout << nilai[4];

// Pemanggilan arraya menggunakan Looping
for(int i = 0; i <5; i++) {
    cout << nilai[i] << endl;
}
```



The screenshot shows a C++ IDE with a terminal window. The terminal has tabs for OUTPUT, DEBUG CONSOLE, PROBLEMS, TERMINAL, PORTS, and C. The terminal output shows the command to run the program and its output.

```
PS C:\Users\Ariq\modul2\lib\output> & .\'guided02.exe'
123451
2
3
4
5
PS C:\Users\Ariq\modul2\lib\output> 
```

### b) Array 2D dan Banyak Dimensi

Array dua dimensi mirip dengan array. Jadi array dua dimensi dapat digunakan untuk menyimpan data dalam bentuk tabel. Terbagi menjadi dua bagian, dimensi pertama dan dimensi kedua. Metode mengakses, mendeklarasikan, menginisialisasi, dan menampilkan data sama dengan array satu dimensi, hanya saja yang digunakan adalah dua indeks. Array multidimensi adalah array dengan banyak indeks, lebih dari dua. Indeks ini menunjukkan ukuran tabel. Array multidimensi lebih sulit divisualisasikan karena jumlah dimensi dalam sebuah array adalah .

```
// ARRAY 2D
int data_nilai[3][4] = {{1,2,3,4}, {4, 6, 7, 8}, {9, 10, 11, 12}};

for (int i = 0; i < 3; i++){
    for (int j = 0; j<4;j++){
        cout << data_nilai[i][j] << " ";
    }
    cout << endl;
}
```

```
// ARRAY BANYAK DIMENSI
int data_rumit[4][6][6];
```

```
PS C:\Users\Ariq\modul2\lib\output> cd 'c:\Users\Ariq\modul2\lib\output'
PS C:\Users\Ariq\modul2\lib\output> & .\'guided02.exe'
1 2 3 4
4 6 7 8
9 10 11 12
PS C:\Users\Ariq\modul2\lib\output> |
```

### 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;
66 }
```

```
PS C:\Users\Ariq\modul2\lib\output> cd 'c:\Users\Ariq\modul2\lib\output'
PS C:\Users\Ariq\modul2\lib\output> & .\'guided02.exe'
Alamat x: 0x8d6bdffa40
Isi px: 0x8d6bdffa40
Isi X: 87
Nilai yang ditunjuk px: 87
Nilai y: 87
```

#### d) Fungsi dan Prosedur

Fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu dengan tujuan utama menyusun program agar mudah dipahami dan dikembangkan. Program ini dibagi menjadi banyak modul kecil. Kedua, dapat mengurangi pengulangan kode (code duplication), sehingga menghemat ukuran program. Biasanya, fungsi memerlukan masukan yang disebut parameter. Masukan ini kemudian diproses oleh fungsi. Hasil akhir dari fungsi tersebut adalah sebuah nilai (nilai kembalian fungsi). Di C sebenarnya tidak ada prosedur, semuanya adalah fungsi, bahkan `main()` pun adalah fungsi. Jadi prosedur dalam C adalah fungsi yang tidak mengembalikan nilai, biasanya diawali dengan kata kunci `void` di depan nama prosedur.

```
5 // FUNGSI
  Codeium: Refactor | Explain | X
6 int penjumlahan(int a, int b){
7     return a + b;
8 }
9
10 // PROSEDUR
   Codeium: Refactor | Explain | X
11 void greet(string name){
12     cout << "Hallo, " << name << "!" << endl;
13 }
```

OUTPUT    DEBUG CONSOLE    PROBLEMS    TERMINAL    PORTS    COMMENTS

```
PS D:\Prsktikum Struktur Data\Praktikum2> cd 'd:\Prsktikum Struktur Data\Praktikum2\GUIDED\output'
PS D:\Prsktikum Struktur Data\Praktikum2\GUIDED\output> & .\'guided02.exe'
hasil 8
Hello, alice!
```

#### 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

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  5  TERMINAL  PORTS  COMMENTS

PS D:\Prsktikum Struktur Data\Praktikum2\KODE> cd 'd:\Prsktikum Struktur Data\Praktikum2\KODE\output'
PS D:\Prsktikum Struktur Data\Praktikum2\KODE\output> & .\'pertemuan2.exe'
Masukkan jumlah elemen data array: 10
Data Array : 0 1 2 3 4 5 6 7 8 9
Nomor Genap : 0, 2, 4, 6, 8
Nomor Ganjil : 1, 3, 5, 7, 9
PS D:\Prsktikum Struktur Data\Praktikum2\KODE\output> 
```

- 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:

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  TERMINAL  PORTS  COMMENTS
PS D:\Prsktikum Struktur Data\Praktikum2\KODE> cd 'd:\Prsktikum Struktur Data\Praktikum2\KODE\output'
PS D:\Prsktikum Struktur Data\Praktikum2\KODE\output> & .\'Soal2.exe'
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][0][1] = 2
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:\Prsktikum Struktur Data\Praktikum2\KODE\output> 
```

- 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:

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  3  TERMINAL  PORTS  COMMENTS

PS D:\Prsktikum Struktur Data\Praktikum2\KODE\output> & .\'soal3.exe'
Masukkan jumlah elemen array: 9
Masukkan 9 elemen array:
12 32 26 22 29 49 79 59 69

MENU
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar
Pilih opsi (1-4): 1
Nilai Maksimum: 79

MENU
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar
Pilih opsi (1-4): 2
Nilai Minimum: 12

MENU
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-rata
4. Keluar
Pilih opsi (1-4): 3
Nilai Rata-rata: 41.8889

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:\Prsktikum Struktur Data\Praktikum2\KODE\output> |
```

## **5. Kesimpulan**

Modul 2 ini mencakup dasar-dasar struktur data dalam bahasa C++, termasuk penggunaan array, pointer, serta penerapan fungsi dan prosedur untuk memanipulasi data secara efisien. Array digunakan untuk menyimpan kumpulan data dengan tipe yang sama, sedangkan pointer berfungsi untuk menyimpan alamat memori dan memungkinkan akses dan modifikasi data secara langsung. Modul ini juga menjelaskan hubungan antara array dan pointer, serta metode meneruskan parameter ke fungsi melalui "panggilan berdasarkan nilai", "panggilan demi penunjuk", dan "panggilan dengan referensi" yang memungkinkan manajemen memori yang fleksibel. Dengan memahami teknik-teknik ini, pemrogram dapat membangun program yang lebih terstruktur, efisien, dan lebih mudah dikelola untuk menyelesaikan berbagai masalah komputer.

