

LAPORAN PRAKTIKUM
Modul 2
“CODE BLOCKS IDE &
PENGENALAN BAHASA C++ (BAGIAN KEDUA)”



Disusun Oleh:
Aji Prasetyo Nugroho - 2211104049
S1SE-07-2

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

A. Tujuan

1. Memahami penggunaan pointer dan alamat memori
2. Mengimplementasikan fungsi dan prosedur dalam program

B. Landasan Teori

Array adalah struktur data yang menyimpan elemen bertipe sama dalam satu variabel dengan indeks tertentu, dan dapat berbentuk satu dimensi, dua dimensi, atau lebih. Pointer, sebaliknya, adalah variabel yang menyimpan alamat memori dari variabel lain, memungkinkan manipulasi langsung terhadap variabel tersebut. Dalam C++, pointer dan array memiliki keterkaitan erat, di mana nama array bertindak sebagai pointer ke elemen pertama. Fungsi dan prosedur di C++ membantu memecah program menjadi bagian yang lebih kecil. Fungsi dapat mengembalikan nilai, sedangkan prosedur (fungsi void) tidak. Parameter dapat diteruskan melalui call by value, call by pointer, atau call by reference untuk menentukan bagaimana data diproses.

C. Guided

a) Array

Array dalam C++ adalah struktur data yang menyimpan sekumpulan elemen bertipe sama di lokasi memori yang berurutan, diakses menggunakan indeks berbasis nol. Array dapat bersifat satu atau multi-dimensi, memiliki ukuran tetap, dan memungkinkan akses cepat ke elemen-elemen, tetapi tidak dapat diubah ukurannya setelah dideklarasikan. Array banyak digunakan karena efisiensinya dalam memori dan kemudahan akses elemen.

Source Code :

```
23 // Array
24 int nilai[5]={1,2,3,4,5};
25 cout << nilai[0];
26 cout << nilai[1];
27 cout << nilai[2];
28 cout << nilai[3];
29 cout << nilai[4];
30
31 ✓ for(int i=0; i<5; i++){
32     cout << nilai[i] << endl;
33 }
```

Output :

```
123451
2
3
4
5
PS D:\Praktikum SD>
```

b) Array 2 dimensi

Array 2 dimensi dalam C++ adalah struktur data yang digunakan untuk menyimpan elemen-elemen dalam bentuk matriks atau tabel, di mana elemen diatur dalam baris dan kolom. Deklarasi array 2 dimensi menggunakan sintaks seperti `int array[baris][kolom]`, dan setiap elemen diakses menggunakan dua indeks: indeks baris dan indeks kolom. Misalnya, `array[2][3]` mengacu pada elemen di baris ke-3 dan kolom ke-4 (indeks dimulai dari 0). Array 2 dimensi cocok digunakan untuk menyimpan data dalam format grid, seperti tabel atau koordinat. Meskipun memiliki kemudahan akses, array 2 dimensi memiliki batasan dalam hal ukuran tetap dan kurang fleksibel dibandingkan struktur data lain seperti `vector`.

Source Code :

```
36 // Array 2 dimensi
37 int nilai[3][4] = {
38     {1,2,3,4},
39     {5,6,7,8},
40     {9,10,11,12}
41 };
42
43 for(int i=0; i<3; i++){
44     for(int j=0; j<4; j++){
45         cout << nilai[i][j] << " ";
46     }
47     cout << endl;
48 }
```

Output :

```
1 2 3 4
5 6 7 8
9 10 11 12
PS D:\Praktikum SD>
```

c) Pointer

Pointer dalam C++ adalah variabel yang menyimpan alamat memori variabel lain, memungkinkan akses dan manipulasi langsung terhadap data di lokasi tersebut. Pointer dideklarasikan dengan `*` dan dapat diakses menggunakan operator dereferensi (`*ptr`). Pointer sering digunakan untuk alokasi memori dinamis, melewati argumen ke fungsi, dan membangun struktur data kompleks. Meski bermanfaat, penggunaan pointer memerlukan kehati-hatian karena kesalahan seperti dereferensi pointer `NULL` dapat menyebabkan crash atau perilaku tak terduga.

Source Code :

```
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
64 getch();
65 }
```

Output :

```
Alamat x= 0x76b03ffa10
Isi px= 0x76b03ffa10
Isi X= 87
Nilai yang ditunjuk px = 87
Nilai y = 87
```

d) Fungsi & procedure

Fungsi dan prosedur dalam C++ adalah blok kode yang dirancang untuk menjalankan tugas tertentu, membantu modularitas dan pengorganisasian program. Fungsi memiliki nilai kembalian (return value) dan dideklarasikan dengan tipe data tertentu, seperti `int` atau `double`, diikuti oleh nama dan parameter, misalnya `int tambah(int a, int b)`. Sebaliknya, prosedur adalah fungsi yang tidak mengembalikan nilai apa pun dan menggunakan kata kunci `void`, seperti `void cetak()`. Fungsi dan prosedur memungkinkan pemrogram memecah program menjadi bagian-bagian kecil yang dapat dipanggil berulang kali, meminimalkan pengulangan kode, dan meningkatkan keterbacaan serta pemeliharaan program.

Source code :

```
6  int penjumlahan(int a, int b){
7      return a + b;
8  }
9
10 void greet(string name){
11     cout << "Hello," << name << "!" << endl;
12 }
13 int main(){
14
15
16     // Fungsi
17     int hasil = penjumlahan(5, 3);
18
19     cout << "Hasil " << hasil << endl;
20
21     greet("alice");
22 }
```

Output :

```
Hasil 8
Hello,alice!
PS D:\Praktikum SD> 
```

D. Unguided

1. 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,
```

Source code :

```
// 1
int n;

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

int dataArray[n];
int genap[n];
int ganjil[n];
int idxGenap = 0, idxGanjil = 0;

cout << "Masukkan elemen-elemen array: ";
for (int i = 0; i < n; ++i) {
    cin >> dataArray[i];

    if (dataArray[i] % 2 == 0) {
        genap[idxGenap++] = dataArray[i];
    } else {
        ganjil[idxGanjil++] = dataArray[i];
    }
}

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

cout << "Nomor Genap : ";
for (int i = 0; i < idxGenap; ++i) {
    cout << genap[i];
    if (i < idxGenap - 1) cout << ", ";
}
cout << endl;

cout << "Nomor Ganjil : ";
for (int i = 0; i < idxGanjil; ++i) {
    cout << ganjil[i];
    if (i < idxGanjil - 1) cout << ", ";
}
cout << endl;

return 0;
```

Output :

```
Masukkan jumlah elemen array: 10
Masukkan elemen-elemen array: 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:\Praktikum SD>
```

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

Source code :

```
// 2
int x, y, z;

cout << "Masukkan ukuran array (dimensi x): ";
cin >> x;
cout << "Masukkan ukuran array (dimensi y): ";
cin >> y;
cout << "Masukkan ukuran array (dimensi z): ";
cin >> z;

int ***array3D = new int**[x];
for (int i = 0; i < x; ++i) {
    array3D[i] = new int*[y];
    for (int j = 0; j < y; ++j) {
        array3D[i][j] = new int[z];
    }
}

cout << "Masukkan elemen-elemen array 3D:" << endl;
for (int i = 0; i < x; ++i) {
    for (int j = 0; j < y; ++j) {
        for (int k = 0; k < z; ++k) {
            cout << "array3D[" << i << "][" << j << "][" << k << "] = ";
            cin >> array3D[i][j][k];
        }
    }
}

cout << "\nArray 3 Dimensi:" << endl;
for (int i = 0; i < x; ++i) {
    for (int j = 0; j < y; ++j) {
        for (int k = 0; k < z; ++k) {
            cout << "array3D[" << i << "][" << j << "][" << k << "] = " << array3D[i][j][k] << endl;
        }
    }
}

for (int i = 0; i < x; ++i) {
    for (int j = 0; j < y; ++j) {
        delete[] array3D[i][j];
    }
    delete[] array3D[i];
}
delete[] array3D;

return 0;
```

Output :

```
Masukkan ukuran array (dimensi x): 2
Masukkan ukuran array (dimensi y): 2
Masukkan ukuran array (dimensi z): 2
Masukkan elemen-elemen array 3D:
array3D[0][0][0] = 1 2 3
array3D[0][0][1] = array3D[0][1][0] = array3D[0][1][1] = 4 5 6
array3D[1][0][0] = array3D[1][0][1] = array3D[1][1][0] = 7 8 9
array3D[1][1][1] =
Array 3 Dimensi:
array3D[0][0][0] = 1
array3D[0][0][1] = 2
array3D[0][1][0] = 3
array3D[0][1][1] = 4
array3D[1][0][0] = 5
array3D[1][0][1] = 6
array3D[1][1][0] = 7
array3D[1][1][1] = 8
PS D:\Praktikum SD>
```


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

Source code

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

int cariMaksimum(int arr[], int n) {
    int maks = arr[0];
    for (int i = 1; i < n; ++i) {
        if (arr[i] > maks) {
            maks = arr[i];
        }
    }
    return maks;
}

int cariMinimum(int arr[], int n) {
    int min = arr[0];
    for (int i = 1; i < n; ++i) {
        if (arr[i] < min) {
            min = arr[i];
        }
    }
    return min;
}

double cariRataRata(int arr[], int n) {
    int total = 0;
    for (int i = 0; i < n; ++i) {
        total += arr[i];
    }
    return static_cast<double>(total) / n;
}

int main() {
    int n, pilihan;

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

    int arr[n];
    cout << "Masukkan elemen-elemen array: ";
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];
    }

    do {
        cout << "\nMenu Pilihan:\n";
        cout << "1. Mencari Nilai Maksimum\n";
        cout << "2. Mencari Nilai Minimum\n";
        cout << "3. Mencari Nilai Rata-Rata\n";
        cout << "4. Keluar\n";
        cout << "Masukkan pilihan: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                cout << "Nilai Maksimum: " << cariMaksimum(arr, n) << endl;
                break;
            case 2:
                cout << "Nilai Minimum: " << cariMinimum(arr, n) << endl;
                break;
            case 3:
                cout << "Nilai Rata-Rata: " << cariRataRata(arr, n) << endl;
                break;
            case 4:
                cout << "Keluar dari program.\n";
                break;
            default:
                cout << "Pilihan tidak valid. Silakan coba lagi.\n";
        }
    } while (pilihan != 4);

    return 0;
}
```

Output :

```
Masukkan jumlah elemen array: 7
Masukkan elemen-elemen array: 1 2 3 4 5 6 7
```

```
Menu Pilihan:
```

1. Mencari Nilai Maksimum
2. Mencari Nilai Minimum
3. Mencari Nilai Rata-Rata
4. Keluar

```
Masukkan pilihan: 1
```

```
Nilai Maksimum: 7
```

```
Menu Pilihan:
```

1. Mencari Nilai Maksimum
2. Mencari Nilai Minimum
3. Mencari Nilai Rata-Rata
4. Keluar

```
Masukkan pilihan: 2
```

```
Nilai Minimum: 1
```

```
Menu Pilihan:
```

1. Mencari Nilai Maksimum
2. Mencari Nilai Minimum
3. Mencari Nilai Rata-Rata
4. Keluar

```
Masukkan pilihan: 3
```

```
Nilai Rata-Rata: 4
```

```
Menu Pilihan:
```

1. Mencari Nilai Maksimum
2. Mencari Nilai Minimum
3. Mencari Nilai Rata-Rata
4. Keluar

```
Masukkan pilihan: 4
```

```
Keluar dari program.
```

```
PS D:\Praktikum SD> █
```