

LAPORAN PRAKTIKUM
Modul 2
“CODEBLOCKS IDE & PENGENALAN BAHASA C++ (BAGIAN 2)”



Disusun Oleh:
Alya Rabani - 2311104076
S1SE-07-B

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

- Memahami penggunaan pointer dan alamat memori
- Mengimplementasikan fungsi dan prosedur dalam program

2. Landasan Teori

❖ Array

Array merupakan struktur data yang digunakan untuk menyimpan sekumpulan data dalam satu tempat. Setiap data dalam array memiliki indeks, sehingga kita akan mudah memprosesnya. Indeks array selalu dimulai dari angka nol (0). Pada teori struktur data array akan bergantung dari banyaknya data yang disimpan di dalamnya. Pada C++, array dapat kita buat dengan cara seperti ini.

// membuat array kosong dengan tipe data integer dan panjang 10

```
int nama_array[10];
```

// membuat array dengan langsung diisi

```
int nama_arr[3] = {0, 3, 2}
```

Cara membuat array hampir sama seperti cara membuat variabel biasa. Bedanya pada array kita harus menentukan panjangnya. Seperti yang sudah kita ketahui array akan menyimpan sekumpulan data dan memberinya nomor indeks agar mudah diakses. Indeks array selalu dimulai dari nol 0. Misalkan kita punya array seperti ini:

```
char huruf[5] = {'a', 'b', 'c', 'd', 'e'};
```

Bagaimana cara mengambil huruf c?

Jawabannya: huruf[2];

Berikut adalah beberapa jenis array:

1. Array 1 dimensi (1D Array)

Merupakan jenis array yang paling sederhana, di mana elemen-elemennya disimpan dalam satu baris.

Contoh: `int arr[5] = {1, 2, 3, 4, 5};` menyimpan lima elemen dalam satu dimensi.

2. Array 2 dimensi (2D Array)

Array ini menyimpan data dalam bentuk tabel, yaitu baris dan kolom.

Contoh: `int arr[2][3] = {{1, 2, 3}, {4, 5, 6}};` memiliki dua baris dan tiga kolom.

3. Array Multidimensi

Array dengan lebih dari dua dimensi. Misalnya, array tiga dimensi bisa dianggap sebagai array dalam bentuk kubus.

Contoh: `int arr[2][3][4];` adalah array tiga dimensi dengan dua lapisan, tiga baris, dan empat kolom.

❖ Pointer

Pointer adalah variabel yang menyimpan alamat memori dari variabel lain yang berisi data. Maka dari itu pointer sangat berhubungan dengan data dan memori. alih-alih menyimpan data langsung, pointer menyimpan lokasi di mana data

tersebut disimpan di dalam memori komputer. Berikut adalah berbagai macam Teknik pointer dalam pemrograman:

- Pointer dan Alamat
Untuk mengambil Alamat variable, Operator & digunakan untuk mengambil alamat memori dari sebuah variabel. Operator * digunakan untuk mengakses atau memodifikasi nilai yang disimpan di alamat memori yang ditunjuk oleh pointer.
- Pointer dan Array
Array dan pointer sangat terkait dalam pemrograman C. Nama array sebenarnya adalah alamat dari elemen pertama array. Pointer dapat digunakan untuk mengakses elemen array dengan aritmetika pointer. Array dapat diakses menggunakan pointer, dan nama array merupakan alamat dari elemen pertama.
- Pointer dan String
String adalah array karakter yang diakhiri dengan karakter null (\0). String dapat diakses menggunakan pointer. Pointer dapat digunakan untuk mengiterasi dan memanipulasi karakter dalam string. Pointer juga dapat digunakan untuk menyimpan alamat pointer lain, memungkinkan manipulasi yang lebih kompleks pada string.

❖ **Prosedur**

Prosedur, dalam konteks pemrograman, adalah sekumpulan instruksi yang dikelompokkan dan diberi nama. Instruksi-instruksi ini dirancang untuk melakukan tugas tertentu. Prosedur ini bisa dipanggil berulang kali kapan pun kita membutuhkan tugas tersebut.

Apakah Prosedur Ada dalam Cpp?

Tidak secara langsung. C++ lebih sering menggunakan istilah 'fungsi' (function) daripada prosedur. Namun, konsepnya sangat mirip. Fungsi dalam C++ juga merupakan blok kode yang dapat dipanggil berkali-kali untuk melakukan tugas tertentu. Perbedaan Prosedur dan Fungsi yaitu, terminologi, Istilah yang lebih umum digunakan adalah fungsi. Lalu, return value fungsi yang dapat mengembalikan nilai (return value), sedangkan prosedur (dalam bahasa pemrograman lain) biasanya tidak. Namun, ini bukan perbedaan yang mutlak, karena fungsi dalam C++ juga bisa dibuat tanpa return value. Dan parameter yang baik fungsi maupun prosedur bisa menerima parameter sebagai input

❖ **Fungsi**

Fungsi adalah sekumpulan kode yang dirancang untuk melakukan tugas tertentu. Bayangkan fungsi sebagai sebuah mesin kecil dalam program Anda. Ketika Anda membutuhkan mesin itu untuk melakukan pekerjaannya, Anda cukup "menyalakannya" (memanggil fungsi tersebut).

Struktur dasar fungsi pada cpp:

```
tipe_data nama_fungsi(parameter1, parameter2, ...) {  
    // Blok kode yang akan dijalankan  
    return nilai_kembali;  
}
```

Tipe data: Jenis data yang akan dikembalikan oleh fungsi. Jika fungsi tidak mengembalikan nilai, gunakan void.

Nama fungsi: Nama yang diberikan untuk fungsi.

Parameter: Variabel yang digunakan untuk menerima input dari pemanggilan fungsi.

Nilai Kembali: Nilai yang akan dikembalikan oleh fungsi.

3. Guided

- a) Program pertama merupakan program yang mencetak nilai-nilai array ke konsol. Pada baris pertama dibuat perintah untuk memberitahu compiler untuk memasukkan file header 'iostream' ke dalam program. File header tersebut berisi fungsi-fungsi input/output standar. Kemudian, digunakan namespace standar 'std'. Lalu, buat fungsi 'main' agar program dapat dijalankan. Setelah itu dibuat sebuah array integer 'nilai' dengan 5 elemen, array nilai diinisialisasi dengan nilai-nilai 1, 2, 3, 4, dan 5. Lalu, gunakan objek 'cout' untuk mencetak nilai elemen pertama array 'nilai' ke layar. Seterusnya akan menggunakan objek 'cout' untuk mencetak nilai 2, 3, 4, dan 5. Jadi, ketika program ini dijalankan, akan mencetak nilai-nilai 1, 2, 3, 4, dan 5 ke layar, dipisahkan oleh spasi. Berikut merupakan contoh program:

```
* guided.cpp > main()
1  #include <iostream>
2  // #include <conio.h>
3  using namespace std;
4  int main()
5  {
6      int nilai[5]={1,2,3,4,5};
7      cout << nilai[0];
8      cout << nilai[1];
9      cout << nilai[2];
10     cout << nilai[3];
11     cout << nilai[4];
12 }
```

Output dari program:

```
PS C:\pertemuan2\outpu
12345
PS C:\pertemuan2\outpu
```

- b) Pada program kedua juga masih menggunakan array sama seperti cara pertama memasukan file header 'iostream, gunakan namespace standar, dan buat fungsi main agar program dapat dijalankan. Kemudian deklarasikan sebuah array integer 'nilai' dengan 5 elemen. Array tersebut diinisialisasi dengan nilai-nilai 1, 2, 3, 4, dan 5. Lalu, dibuat sebuah perulangan 'for' yang akan dijalankan sebanyak 5 kali. Variabel 'i' akan diinisialisasi dengan nilai 0, dan akan bertambah 1 setiap kali perulangan dijalankan. Perulangan akan berhenti Ketika 'i' mencapai nilai 5. Setelah itu, buat objek 'cout' untuk mencetak nilai elemen array 'nilai' ke layar. Nilai elemen array 'nilai' yang dicetak adalah nilai yang sesuai dengan indeks 'i'. Misalnya, pada iterasi pertama, 'i' bernilai 0, maka nilai elemen pertama array 'nilai' yang dicetak adalah 1. Pada iterasi kedua, 'i' bernilai 1, maka nilai elemen kedua array 'nilai' yang dicetak adalah 2, dan seterusnya. Setelah perulangan selesai, gunakan 'endl' untuk membuat baris baru setelah mencetak nilai. Jadi, setiap nilai akan dicetak di baris yang berbeda. Jadi, ketika program ini dijalankan, akan mengeluarkan hasil nilai-nilai 1, 2, 3, 4, dan 5 ke layar, masing-masing di baris yang berbeda. Berikut contoh programnya:

```

int main()
{
    int nilai[5]{1,2,3,4,5};

    for(int i=0; i<5; i++){
        cout << nilai[i] << endl;
    }
}

```

Output dari program:

```

PS C:\pertemuan2\ou
1
2
3
4
5

```

- c) Pada program ketiga merupakan mendeklarasikan array 2 dimensi dengan memasukan header 'iostream', namespace, dan fungsi main. Kemudian, dibuat sebuah perintah yang mendeklarasikan sebuah array dua dimensi 'nilai' dengan ukuran 3x4. Array nilai memiliki 3 baris dan 4 kolom. Nilai-nilai dalam array nilai diinisialisasi dengan nilai-nilai yang tercantum dalam kurung kurawal. Pada baris selanjutnya dibuat sebuah perulangan 'for' yang akan dijalankan sebanyak 3 kali. Variabel 'i' akan diinisialisasi dengan nilai 0, dan akan bertambah 1 setiap kali perulangan dijalankan. Perulangan akan berhenti ketika i mencapai nilai 3. Digunakan lagi perulangan 'for' yang akan dijalankan sebanyak 4 kali. Variabel 'j' akan diinisialisasi dengan nilai 0, dan akan bertambah 1 setiap kali perulangan dijalankan. Perulangan akan berhenti Ketika 'j' mencapai nilai 4. Jika perulangan sudah selesai, digunakan objek 'cout' untuk mencetak nilai elemen array 'nilai' ke layar. Nilai elemen array 'nilai' yang dicetak adalah nilai yang sesuai dengan indeks 'i' dan 'j'. Misalnya, pada iterasi pertama, 'i' bernilai 0 dan 'j' bernilai 0, maka nilai elemen pertama array 'nilai' yang dicetak adalah 1. Kemudian, panggil 'cout' dan 'endl' untuk membuat baris baru setelah mencetak nilai. Berikut contoh programnya

```

int main()
{
    int nilai[3][4] = {
        {1,2,3,4},
        {5,6,7,8},
        {9,10,11,12}
    };
    for(int i=0; i<3; i++){
        for(int j=0; j<4; j++){
            cout << nilai[i][j] << " ";
        }
        cout << endl;
    }
}

```

Output dari program:

```
PS C:\pertemuan2\outp
1 2 3 4
5 6 7 8
9 10 11 12
PS C:\pertemuan2\outp
```

- d) Pada program ke empat dibuat sebuah program menggunakan pointer untuk menunjuk alamat memori sebuah variabel dan mencetak nilai-nilai yang terkait dengan pointer dan variabel tersebut. Hampir sama seperti biasanya digunakan header 'iostream', 'namespace' standar, dan juga fungsi 'main'. Tetapi ada satu header yang harus ditambahkan yaitu header 'conio.h' yang berisi fungsi-fungsi input/output yang lebih lanjut, seperti 'getch()' yang digunakan untuk menghentikan program sementara. Jika sudah, dibuat beberapa deklarasi seperti dua variabel integer 'x' dan 'y'. lalu deklarasi sebuah pointer integer 'px' dimana pointer 'px' dapat menunjuk ke alamat memori sebuah variabel integer. Setelah itu inisialisasikan pointer 'px' dengan alamat memori variable 'x'. Operator '&' digunakan untuk mendapatkan alamat memori sebuah variable. Lalu, inisialisasikan lagi variable 'y' dengan nilai yang ditunjuk oleh pointer 'px'. Operator '*' digunakan untuk mendapatkan nilai yang ditunjuk oleh pointer. Kemudian dibuat perintah yang menggunakan objek 'cout' untuk mencetak setiap variable yang sudah diinputkan dan tambahkan 'endl' agar mencetak yang ditunjuk oleh pointer ke layar. Jika semua sudah dimasukan, gunakan fungsi 'getch()' untuk menghentikan program sementara dan menunggu input dari pengguna. Berikut contoh dari program:

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int x,y;
    int *px;
    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;
    getch();
}
```

Output dari program:

```
PS C:\pertemuan2\output> & .\'guided
Alamat x= 0xf1525ffb40
Isi px= 0xf1525ffb40
Isi X= -1327557720
Nilai yang ditunjuk px= -1327557720
Nilai y= -1327557720
```

- e) Pada program kelima merupakan sebuah prosedur yang dapat dipanggil berulang kali dari berbagai bagian program untuk melakukan tugas tertentu. Sama seperti pointer, prosedur juga menggunakan dua header yaitu 'iostream' dan 'conio.h', lalu 'namespace'. Setelah itu, dibuat perintah untuk mendeklarasikan sebuah prosedur 'greet' yang tidak mengembalikan nilai ('void'). Prosedur 'greet' memiliki satu parameter 'name' dengan tipe data 'string'. Lalu, dibuat objek 'cout' untuk

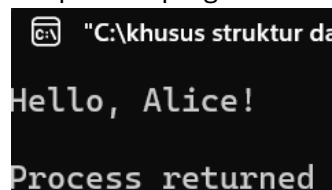
mencetak pesan "Hello, " diikuti dengan nilai parameter 'name' dan diakhiri dengan tanda seru (!) ke layar. Tutup prosedur greet dan deklarasikan fungsi 'main' agar program dapat dijalankan. Panggil prosedur 'greet' dengan parameter "Alice". Kembalikan nilai 0 ke sistem operasi, menunjukkan bahwa program telah berjalan dengan sukses. Jika sudah tutup fungsi 'main'. Berikut adalah contoh programnya:

```
// prosedur
#include <iostream>
#include <conio.h>
using namespace std;

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

int main(){
    greet("Alice");
    return 0;
}
```

Output dari program:



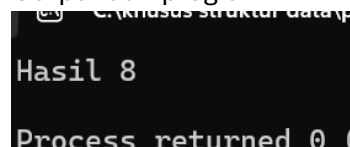
- f) Pada program keenam ini merupakan fungsi 'penjumlahan' yang melakukan penjumlahan antara dua nilai dan hasilnya disimpan dalam variabel 'hasil'. Hasilnya dicetak menggunakan objek 'cout'. Pertama-tama gunakan header 'iostream', lalu buat deklarasi 'penjumlahan' yang menerima dua parameter 'a' dan 'b' dengan tipe data 'int'. Kemudian kembalikan hasil penjumlahan antara **a** dan **b** ke program utama. Deklarasikan fungsi 'main' agar program bisa berjalan dan panggil fungsi 'penjumlahan' dengan parameter '5' dan '3', dan menyimpan hasilnya dalam variabel 'hasil' dengan tipe data 'int'. Gunakan objek 'cout' untuk mencetak string "Hasil " diikuti dengan nilai 'hasil' ke layar. Perintah 'std::endl' digunakan untuk membuat baris baru setelah mencetak output. Kembalikan nilai 0 ke sistem operasi, menunjukkan bahwa program telah berjalan dengan sukses. Berikut adalah contoh programnya:

```
#include <iostream>

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

int main(){
    int hasil = penjumlahan(5, 3);
    std::cout << "Hasil " << hasil << std::endl;
    return 0;
}
```

Output dari program:



4. Unguided

- 1) Program menginputkan data array lalu dipisahkan mana yang ganjil dan mana yang genap.

```
1 //1
2 #include <iostream>
3 #include <vector>
4
5 using namespace std;
6
7 int main() {
8     int n;
9     cout << "Masukkan jumlah data: ";
10    cin >> n;
11
12    vector<int> data(n);
13    vector<int> genap;
14    vector<int> ganjil;
15
16    cout << "Masukkan " << n << " data: ";
17    for (int i = 0; i < n; ++i) {
18        cin >> data[i];
19        if (data[i] % 2 == 0) {
20            genap.push_back(data[i]);
21        } else {
22            ganjil.push_back(data[i]);
23        }
24    }
25
26    cout << "Data Array: ";
27    for (int i = 0; i < n; ++i) {
28        cout << data[i] << " ";
29    }
30    cout << endl;
31
32    cout << "Nomor Genap: ";
33    for (int i = 0; i < genap.size(); ++i) {
34        cout << genap[i];
35        if (i < genap.size() - 1) {
36            cout << ", ";
37        }
38    }
39    cout << ", " << endl;
40
41    cout << "Nomor Ganjil: ";
42    for (int i = 0; i < ganjil.size(); ++i) {
43        cout << ganjil[i];
44        if (i < ganjil.size() - 1) {
45            cout << ", ";
46        }
47    }
48    cout << ", " << endl;
49
50    return 0;
51 }
```

Output program:

```
PS C:\pertemuan2\output> g++ 4-1-unguided.exe
Masukkan jumlah data: 12
Masukkan 12 data: 22 65 44 12 24 99 77 66 35 45 35 61
Data Array: 22 65 44 12 24 99 77 66 35 45 35 61
Nomor Genap: 22, 44, 12, 24, 66,
Nomor Ganjil: 65, 99, 77, 35, 45, 35, 61,
PS C:\pertemuan2\output>
```


- 2) Program input array 3 dimensi yang diinputkan oleh user

```
1 #include <iostream>
2 #include <vector>
3 #include <sstream>
4
5 using namespace std;
6
7 int main() {
8     int x, y, z;
9
10    // Menerima input dimensi array dari user
11    cout << "Masukkan dimensi array (x y z): ";
12    cin >> x >> y >> z;
13
14    // Membuat array tiga dimensi dengan ukuran dinamis
15    vector<vector<vector<int>>> arr(x, vector<vector<int>>(y, vector<int>(z)));
16
17    // Menerima input elemen array dari user
18    cout << "Masukkan elemen array:\n";
19    for (int i = 0; i < x; ++i) {
20        for (int j = 0; j < y; ++j) {
21            for (int k = 0; k < z; ++k) {
22                cin >> arr[i][j][k];
23            }
24        }
25    }
26
27    // Menampilkan isi array
28    cout << "\nIsi array tiga dimensi:\n";
29    for (int i = 0; i < x; ++i) {
30        for (int j = 0; j < y; ++j) {
31            for (int k = 0; k < z; ++k) {
32                cout << arr[i][j][k] << " ";
33            }
34            cout << endl;
35        }
36        cout << endl;
37    }
38
39    return 0;
40 }
```

Output program:

```
Masukkan dimensi array (x y z): 2 3 2
Masukkan elemen array:
1 2 3
4 5 6
7 8 9
10 11 12

Isi array tiga dimensi:
1 2
3 4
5 6

7 8
9 10
11 12
```

- 3) Program menu untuk mencari nilai Maksimum, Minimum dan Nilai rata – rata dari suatu array

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  double cari_maksimum(vector<int> arr) {
8      return *max_element(arr.begin(), arr.end());
9  }
10
11 double cari_minimum(vector<int> arr) {
12     return *min_element(arr.begin(), arr.end());
13 }
14
15 double cari_rata_rata(vector<int> arr) {
16     double sum = 0;
17     for (int i = 0; i < arr.size(); i++) {
18         sum += arr[i];
19     }
20     return sum / arr.size();
21 }
22
23 int main() {
24     cout << "Program Menu" << endl;
25     cout << "1. Cari Nilai Maksimum" << endl;
26     cout << "2. Cari Nilai Minimum" << endl;
27     cout << "3. Cari Nilai Rata-Rata" << endl;
28     cout << "4. Keluar" << endl;
29
30     while (true) {
31         int pilihan;
32         cout << "Masukkan pilihan: ";
33         cin >> pilihan;
34
35         if (pilihan == 1) {
36             vector<int> arr;
37             int num;
38             cout << "Masukkan array (pisahkan dengan spasi): ";
39             while (cin >> num) {
40                 arr.push_back(num);
41                 if (cin.peek() == '\n') break;
42             }
43             cout << "Nilai Maksimum: " << cari_maksimum(arr) << endl;
44         } else if (pilihan == 2) {
45             vector<int> arr;
46             int num;
47             cout << "Masukkan array (pisahkan dengan spasi): ";
48             while (cin >> num) {
49                 arr.push_back(num);
50                 if (cin.peek() == '\n') break;
51             }
52             cout << "Nilai Minimum: " << cari_minimum(arr) << endl;
53         } else if (pilihan == 3) {
54             vector<int> arr;
55             int num;
56             cout << "Masukkan array (pisahkan dengan spasi): ";
57             while (cin >> num) {
58                 arr.push_back(num);
59                 if (cin.peek() == '\n') break;
60             }
61             cout << "Nilai Rata-Rata: " << cari_rata_rata(arr) << endl;
62         } else if (pilihan == 4) {
63             cout << "Keluar dari program" << endl;
64             break;
65         } else {
66             cout << "Pilihan tidak valid. Silakan coba lagi." << endl;
67         }
68     }
69
70     return 0;
71 }

```

Output dari program:

```
Program Menu
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-Rata
4. Keluar
Masukkan pilihan: 1
Masukkan array (pisahkan dengan spasi): 88 28 76 19
Nilai Maksimum: 88
Masukkan pilihan: 2
Masukkan array (pisahkan dengan spasi): 19 40 23 20 9
Nilai Minimum: 9
Masukkan pilihan: 3
Masukkan array (pisahkan dengan spasi): 40 23 89 51 28
Nilai Rata-Rata: 46.2
```

5. Kesimpulan

Kesimpulan dari laporan praktikum Modul 2 ini menekankan pada pemahaman tentang penggunaan pointer, array, dan fungsi dalam pemrograman C++. Mahasiswa diajarkan untuk menggunakan array untuk menyimpan sekumpulan data, baik dalam satu dimensi maupun multidimensi, serta cara mengakses data tersebut menggunakan indeks. Memahami konsep pointer yang berfungsi untuk menyimpan alamat memori dari suatu variabel dan bagaimana pointer dapat digunakan dalam manipulasi data. Memahami cara kerja fungsi dalam C++ yang digunakan untuk menjalankan tugas-tugas spesifik dan dapat dipanggil berulang kali.

Mengimplementasikan prosedur untuk melakukan operasi tertentu dalam program tanpa mengembalikan nilai, serta perbedaan antara prosedur dan fungsi dalam konteks C++. Praktikum ini berhasil memberikan gambaran yang jelas tentang dasar-dasar struktur data, penggunaan pointer, dan konsep fungsi, yang penting dalam pengembangan program menggunakan bahasa C++.