

LAPORAN PRAKTIKUM
Modul II
“02_Pengenalan_CPP_Bagian_2”



Disusun Oleh:
Dewi Atika Muthi -2211104042
SE-07-02

Dosen:
Wahyu Andi Saputra

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

Tujuan dari praktikum ini adalah untuk:

- Memahami konsep dasar array dan pointer.** Mahasiswa diharapkan dapat memahami bagaimana array dan pointer bekerja dalam penyimpanan serta manipulasi data.
- Mengimplementasikan fungsi dan prosedur.** Praktikum ini juga bertujuan untuk mengembangkan pemahaman tentang bagaimana membuat dan menggunakan fungsi serta prosedur dalam pemrograman C++.
- Meningkatkan keterampilan pemrograman dengan string dan manipulasi data teks.** Mahasiswa akan belajar cara memanipulasi string dengan efisien menggunakan array karakter dan pointer.

2. Landasan Teori

Beberapa konsep penting yang mendasari praktikum ini adalah:

- Array:** Array digunakan untuk menyimpan beberapa data dalam satu variabel. Setiap elemen array dapat diakses menggunakan indeks numerik, dengan elemen pertama berada pada indeks 0.
- Pointer:** Pointer menyimpan alamat memori suatu variabel, memungkinkan manipulasi langsung terhadap data yang disimpan di lokasi tersebut. Dengan pointer, kita bisa mengakses elemen array lebih efisien.
- String:** Dalam C++, string diimplementasikan sebagai array karakter yang diakhiri dengan karakter null (`\0`), memungkinkan berbagai operasi teks seperti penggabungan, perbandingan, dan pemotongan.
- Fungsi dan Prosedur:** Fungsi adalah blok kode yang mengembalikan nilai setelah menyelesaikan tugasnya, sementara prosedur tidak mengembalikan nilai apapun. Penggunaan fungsi/prosedur memudahkan pemrograman modular dan mencegah pengulangan kode.

3. Guided

A. Array

1) Array Satu Dimensi

Array satu dimensi adalah array yang hanya terdiri dari satu larik data saja. Cara pendeklarasian array satu dimensi:

`tipe_data nama_var[ukuran]`

Contoh:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int nilai[5] = { 3, 4, 5, 6, 7 };
8      cout << nilai[2];
9  }
```

Output:

```
C:\StrukturData\prak3\bin\De x + v
5
Process returned 0 (0x0)   execution time : 0.327 s
Press any key to continue.
```

Penjelasan:

Indeks array **dimulai dari 0**, sehingga `nilai[2]` yang mengarah pada nilai elemen array nilai pada indeks ke-2, artinya merujuk pada elemen ketiga dalam array, yaitu angka 5. Maka yang akan di tampilkan dalam program adalah 5

2) Array Dua Dimensi

Bentuk **array dua dimensi** ini 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, inialisasi, dan menampilkan data sama dengan array satu dimensi, hanya saja indeks yang digunakan ada dua.

Contoh:

```
int data_nilai[4][3];
nilai[2][0] = 10;
```

	0	1	2
0			
1			
2	10		
3			

Gambar 2-1 Ilustrasi Array Dua Dimensi

Misal penulisan dalam program:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int nilai[3][4] = {
8          {1, 2, 3, 4},
9          {5, 6, 7, 8},
10         {9, 10, 11, 12}
11     };
12
13     // Loop untuk mencetak elemen array
14     for (int i = 0; i < 3; i++) {
15         for (int j = 0; j < 4; j++) {
16             cout << nilai[i][j] << " ";
17         }
18         cout << endl;
19     }
20
21     return 0;
22 }
23
```

Output:

```
C:\StrukturData\modul2_KAG >
1 2 3 4
5 6 7 8
9 10 11 12

Process returned 0 (0x0)   execution time : 0.177 s
Press any key to continue.
```

Penjelasan:

- Nilai[3][4] adalah array dua dimensi dengan 3 baris dan 4 kolom,

```
int nilai[3][4] = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12}
};
```

yang nilainya diisi seperti:

- Program menggunakan **nested loop (for)** untuk mengakses setiap elemen dalam array tersebut.
- Loop pertama berjalan dari 0 hingga 2 (3 iterasi untuk 3 baris), dan loop kedua berjalan dari 0 hingga 3 (4 iterasi untuk 4 kolom).
- Untuk setiap iterasi loop, program mencetak elemen array yang sesuai, diikuti dengan spasi. Setelah selesai mencetak satu baris (loop kedua selesai), program mencetak baris baru (**endl**).

B. Pointer

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.

Cara pendeklarasian variabel pointer adalah sebagai berikut:

```
type *nama_variabel;
```

Contoh:

```
int *p_int;
```

`p_int` merupakan variabel pointer yang menunjuk ke data bertipe `int`

Agar suatu pointer menunjuk ke variabel lain, mula-mula pointer harus diisi dengan alamat memori yang ditunjuk.

```
p_int = &j;
```

Untuk mendapatkan nilai dari variabel yang ditunjuk pointer, gunakan tanda `*` di depan nama variabel pointer

Contoh menggunakan pointer:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int x, y;
7     int *px;
8     x = 87;
9     px = &x;
10    y = *px;
11
12    cout << "Alamat x = " << &x << endl;
13    cout << "Isi px = " << px << endl;
14    cout << "Isi x = " << x << endl;
15    cout << "Nilai yang ditunjuk px = " << *px << endl;
16    cout << "Nilai y = " << y << endl;
17
18 }
```

Output:

```
Alamat x = 0x61fe10
Isi px = 0x61fe10
Isi x = 87
Nilai yang ditunjuk px = 87
Nilai y = 87

Process returned 0 (0x0)   execution time : 0.125 s
Press any key to continue.
```

Penjelasan:

Deklarasi variabel:

- `int x, y;` mendeklarasikan variabel integer x dan y.
- `int *px;` mendeklarasikan pointer px yang akan menunjuk ke tipe data integer.

Inisialisasi:

- `x = 87;` memberikan nilai 87 ke variabel x.
- `px = &x;` memberikan alamat memori dari x ke pointer px.
- `y = *px;` memberikan nilai yang ditunjuk oleh px (yaitu nilai dari x, yang 87) ke variabel y.

Cetak hasil:

- `cout << "Alamat x " << &x << endl;` akan mencetak alamat memori di mana variabel x disimpan.
- `cout << "Isi px = " << px << endl;` akan mencetak alamat yang disimpan di dalam pointer px, yaitu alamat dari x.
- `cout << "Isi x = " << x << endl;` akan mencetak nilai dari variabel x, yaitu 87.
- `cout << "Nilai yang ditunjuk px = " << *px << endl;` akan mencetak nilai yang ditunjuk oleh px, yaitu 87 (nilai dari x).
- `cout << "Nilai y = " << y << endl;` akan mencetak nilai dari variabel y, yang juga 87.

C. Fungsi

Fungsi merupakan blok dari kode yang dirancang untuk melaksanakan tugas khusus dengan tujuan:

1. Program menjadi terstruktur, sehingga mudah dipahami dan mudah dikembangkan. Program dibagi menjadi beberapa modul yang kecil.

2. 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). Bentuk umum sebuah fungsi:

```
tipe_keluaran nama_fungsi(daftar_parameter) {  
    blok pernyataan fungsi ;  
}
```

Misal kita contohkan pada code di bawah ini:

```
1  #include <iostream>  
2  #include <string>  
3  
4  using namespace std;  
5  
6  int penjumlahan(int a, int b) {  
7      return a + b;  
8  }  
9  
10 void greet(string name) { // Ganti String dengan string  
11     cout << "Hello, " << name << "!" << endl;  
12 }  
13  
14 int main() {  
15     int hasil = penjumlahan(5, 6);  
16     cout << "Hasil: " << hasil << endl;  
17  
18     greet("Tika");  
19  
20     return 0; //  
21 }  
22
```

Output:

```
C:\StrukturData\madul2_KAC  
Hasil: 11  
Hello, Tika!  
Process returned 0 (0x0) execution time : 0.125 s  
Press any key to continue.
```

Penjelasan:

Program ini bertujuan untuk melakukan dua hal:

1. Menghitung jumlah dua bilangan bulat menggunakan fungsi `penjumlahan`.
2. Menampilkan pesan sapaan kepada pengguna menggunakan fungsi `greet`.

Deskripsi fungsi:

- Fungsi `penjumlahan(int a, int b)`: Fungsi ini menerima dua argumen integer, `a` dan `b`, kemudian mengembalikan hasil penjumlahannya.
- Fungsi `greet(string name)`: Fungsi ini menerima satu argumen bertipe string, kemudian mencetak sapaan ke layar dengan format `"Hello, [name]!"`.
- Fungsi `main()`: Di dalam fungsi utama:

- Program memanggil fungsi penjumlahan dengan argumen 5 dan 6, menyimpan hasilnya dalam variabel hasil, lalu mencetaknya.
- Program memanggil fungsi greet dengan argumen "Tika", mencetak sapaan kepada pengguna.

4. Unguided

Soal

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,
```

Jawaban:

Source Code:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int arr[10];
6
7      cout << "Masukkan 10 angka:\n";
8      for(int i = 0; i < 10; i++) {
9          cin >> arr[i];
10     }
11
12     cout << "Data Array : ";
13     for(int i = 0; i < 10; i++) {
14         cout << arr[i] << " ";
15     }
16
17     cout << "\nNomor Genap : ";
18     for(int i = 0; i < 10; i++) {
19         if(arr[i] % 2 == 0) {
20             cout << arr[i] << ", ";
21         }
22     }
23
24     cout << "\nNomor Ganjil : ";
25     for(int i = 0; i < 10; i++) {
26         if(arr[i] % 2 != 0) {
27             cout << arr[i] << ", ";
28         }
29     }
30
31     return 0;
32 }
33
```

Output:

```
Masukkan 10 angka:
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,
Process returned 0 (0x0)   execution time : 6.668 s
Press any key to continue.
```

Penjelasan Program:

Deklarasi Array:

- `int arr[10];` : Mendeklarasikan array arr yang dapat menyimpan 10 elemen bertipe int. Array ini akan digunakan untuk menyimpan angka-angka yang dimasukkan oleh pengguna.

Input Pengguna:

- `cout << "Masukkan 10 angka:\n";` : Meminta pengguna memasukkan angka.
- `for(int i = 0; i < 10; i++) { cin >> arr[i]; }` : Menggunakan loop for untuk membaca 10 angka dari pengguna dan menyimpannya ke dalam array arr.

Menampilkan Data Array:

- `cout << "Data Array : ";` : Menampilkan judul untuk data array.
- `for(int i = 0; i < 10; i++) { cout << arr[i] << " "; }` : Menggunakan loop for lain untuk mencetak semua elemen dalam array arr, dipisahkan dengan spasi.

Menampilkan **Angka Genap**:

- `cout << "\nNomor Genap : ";` : Menampilkan judul untuk angka genap.
- `for(int i = 0; i < 10; i++) { if(arr[i] % 2 == 0) { cout << arr[i] << ", "; } }` : Loop ini memeriksa setiap elemen dalam array. Jika elemen tersebut genap (**divisible by 2**), maka elemen tersebut dicetak.

Menampilkan **Angka Ganjil**:

- `cout << "\nNomor Ganjil : ";` : Menampilkan judul untuk angka ganjil.
- `for(int i = 0; i < 10; i++) { if(arr[i] % 2 != 0) { cout << arr[i] << ", "; } }` : Sama seperti sebelumnya, loop ini memeriksa setiap elemen dalam array. Jika elemen tersebut ganjil (**not divisible by 2**), maka elemen tersebut dicetak.

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

Jawaban:

Source Code:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x, y, z;
6      cout << "Masukkan ukuran array 3D (x y z): ";
7      cin >> x >> y >> z;
8
9      int arr[10][10][10]; // Ukuran maksimum 10x10x10
10
11     cout << "Masukkan elemen array:\n";
12     for(int i = 0; i < x; i++)
13         for(int j = 0; j < y; j++)
14             for(int k = 0; k < z; k++)
15                 cin >> arr[i][j][k];
16
17     cout << "Array 3D yang dimasukkan:\n";
18     for(int i = 0; i < x; i++) {
19         for(int j = 0; j < y; j++) {
20             for(int k = 0; k < z; k++)
21                 cout << arr[i][j][k] << " ";
22             cout << endl;
23         }
24         cout << endl;
25     }
26
27     return 0;
28 }
```


Output:

```
C:\StrukturData\unguided2\lb x + v
Masukkan ukuran array 3D (x y z): 2 2 2
Masukkan elemen array:
1 2
2 3
3 4
4 5
Array 3D yang dimasukkan:
1 2
2 3

3 4
4 5

Process returned 0 (0x0)   execution time : 9.692 s
Press any key to continue.
```

Penjelasan Program:

Deklarasi Variabel:

- `int x, y, z;` Deklarasi tiga variabel int untuk menyimpan ukuran dimensi array 3D.
- `int arr[10][10][10];` Mendeklarasikan array 3D dengan ukuran maksimum 10x10x10. Ini berarti array dapat menyimpan hingga 1000 elemen.

Input **Ukuran Array:**

- `cout << "Masukkan ukuran array 3D (x y z): ";` Mengeluarkan pesan untuk meminta pengguna memasukkan ukuran dimensi array 3D.
- `cin >> x >> y >> z;` Menerima input tiga angka yang mewakili ukuran dimensi x, y, dan z.

Input **Elemen Array:**

- `cout << "Masukkan elemen array:\n";` Mengeluarkan pesan untuk meminta pengguna memasukkan elemen array.
- Tiga loop bersarang digunakan untuk membaca elemen dari array 3D, di mana *i*, *j*, dan *k* masing-masing mewakili indeks untuk dimensi x, y, dan z.

Menampilkan Array 3D:

- `cout << "Array 3D yang dimasukkan:\n";` Mengeluarkan judul untuk menampilkan array 3D.
- Tiga loop bersarang digunakan untuk mencetak elemen dari array 3D, menampilkan setiap elemen dalam satu baris, dan menambahkan baris baru untuk setiap dimensi kedua (*y*).

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

Jawaban:

Source Code:

```

#include <iostream>
using namespace std;

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

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

float rata(int arr[], int n) {
    float sum = 0;
    for(int i = 0; i < n; i++) {
        sum += arr[i];
    }
    return sum / n;
}

int main() {
    int n, choice;
    int arr[100];

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

    if (n <= 0 || n > 100) {
        cout << "Jumlah elemen harus antara 1 dan 100.\n";
        return 1;
    }

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

    do {
        cout << "\nMenu:\n";
        cout << "1. Cari Nilai Maksimum\n";
        cout << "2. Cari Nilai Minimum\n";
        cout << "3. Hitung Nilai Rata-rata\n";
        cout << "4. Keluar\n";
        cout << "Pilih menu: ";
        cin >> choice;

        switch(choice) {
            case 1:
                cout << "Nilai Maksimum: " << maks(arr, n) << endl;
                break;
            case 2:
                cout << "Nilai Minimum: " << min(arr, n) << endl;
                break;
            case 3:
                cout << "Nilai Rata-rata: " << rata(arr, n) << endl;
                break;
            case 4:
                cout << "Terima kasih!\n";
                break;
            default:
                cout << "Pilihan tidak valid. Silakan coba lagi.\n";
        }
    } while(choice != 4);

    return 0;
}

```

Output:

```

Masukkan jumlah elemen array: 5
Masukkan elemen array:
12 3 4 9 16

Menu:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Hitung Nilai Rata-rata
4. Keluar
Pilih menu: 1
Nilai Maksimum: 16

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

Menu:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Hitung Nilai Rata-rata
4. Keluar
Pilih menu: 3
Nilai Rata-rata: 8.8

Menu:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Hitung Nilai Rata-rata
4. Keluar
Pilih menu: 4
Terima kasih!

Process returned 0 (0x0)   execution time : 27.147 s
Press any key to continue.

```

Penjelasan Program:

Fungsi maks, min, dan rata:

- **Fungsi maks:**

Fungsi ini mencari nilai maksimum dalam array arr yang berisi n elemen.

- `if(arr[i] > max)`: Jika elemen saat ini (`arr[i]`) lebih besar dari nilai maksimum yang telah ditemukan (`max`), maka `max` diperbarui dengan nilai tersebut.

- **Fungsi min:**

Fungsi ini mencari nilai minimum dalam array arr.

- `if(arr[i] < min)`: Jika elemen saat ini (`arr[i]`) lebih kecil dari nilai minimum yang telah ditemukan (`min`), maka `min` diperbarui dengan nilai tersebut.

- **Fungsi rata:**

Fungsi ini menghitung rata-rata dari elemen dalam array arr.

- Rata-rata dihitung dengan menjumlahkan semua elemen (`sum`) dan membagi dengan jumlah elemen (`n`).

- **Fungsi main (input elemen):**

- `cin >> arr[i]`; Pengguna diminta untuk memasukkan jumlah elemen yang ingin disimpan di dalam array, kemudian memasukkan elemen-elemen tersebut.

- Kondisi pada Menu Interaksi:

Menggunakan `switch` untuk memproses pilihan:

- Kasus 1: Memanggil fungsi maks dan menampilkan nilai maksimum.
- Kasus 2: Memanggil fungsi min dan menampilkan nilai minimum.
- Kasus 3: Memanggil fungsi rata dan menampilkan nilai rata-rata.
- Kasus 4: Program mengucapkan terima kasih dan keluar.
- Default: Jika pilihan tidak valid, program menampilkan pesan kesalahan.

5. Kesimpulan

Dari praktikum ini, mahasiswa memperoleh pemahaman yang lebih dalam tentang penggunaan array dan pointer, yang memungkinkan penyimpanan data bertipe sama secara efisien serta memberikan akses yang fleksibel ke memori. Selain itu, mahasiswa juga belajar tentang implementasi fungsi dan prosedur, yang memudahkan pemrograman modular dan pengelolaan tugas berulang.

Terakhir, pemahaman mengenai manajemen string dengan array karakter menjadi penting untuk manipulasi data berbasis teks dalam bahasa C++. Secara keseluruhan, praktikum ini memberikan dasar yang kuat dalam konsep-konsep pemrograman dasar di C++.