

**I. COVER**

**LAPORAN PRAKTIKUM**  
**Modul 2**  
**Pemograman lanjut C++ dan Pemahaman Array,Pointer Fungsi,Prosedur**



**Disusun Oleh:**  
**Haza Zaidan Zidna Fann**  
**(2311104056)**

**Kelas:**  
**SISE 07 02**

**Dosen :**  
**Wahyu Andi Saputra**  
**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**PURWOKERTO**  
**2024**

## II. TUJUAN

Mampu mengimplementasikan konsep-konsep penting dalam pemrograman C++ Khususnya tentang Penggunaan pointer dan alamat memori serta mengimplementasikan fungsi dan prosedur dalam program

## III. Landasan Teori

Array : digunakan untuk menyimpan data dengan tipe yang sama termasuk array satu dimensi dua dimensi dan berdimensi banyak

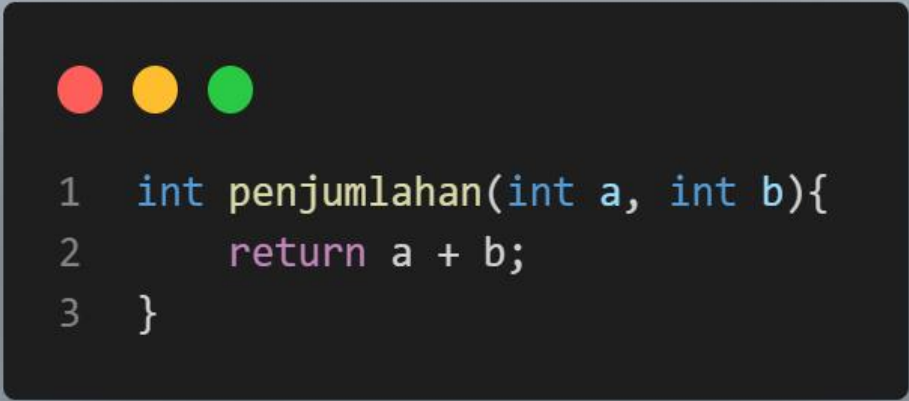
Pointer : menjelaskan cara kerja pointer dalam mengakses alamat memori pointer dalam mengakses alamat memori, pointer, dengan array dan keterhubungan antara pointer dan string

Fungsi : menjelaskan struktur fungsi dalam C++ utk meningkatkan keteraturan kode dan mengurangi duplikasi

Prosedur : sama dengan fungsi tetapi tanpa nilai balik digunakan untuk menyederhanakan proses pemrograman

## IV. GUIDED

### 1. Fungsi untuk penjumlahan dua bilangan :



```
1  int penjumlahan(int a, int b){  
2      return a + b;  
3  }
```

Fungsi `penjumlahan(int a, int b)` adalah sebuah blok kode yang dirancang khusus untuk menjumlahkan dua bilangan bulat.


`int`: Menunjukkan bahwa fungsi ini akan menghasilkan nilai berupa bilangan bulat.

`penjumlahan`: Adalah nama yang diberikan kepada fungsi ini.

`(int a, int b)`: Menentukan bahwa fungsi ini menerima dua input berupa bilangan bulat yang akan dinamakan `a` dan `b` di dalam fungsi.

return a + b;; Menjumlahkan nilai a dan b, lalu mengembalikan hasilnya.

## 2. Fungsi untuk memanggil sapaan :



```
1 void greet(string name){
2     cout << "Hello, " << name << "!" << endl;
3 }
```

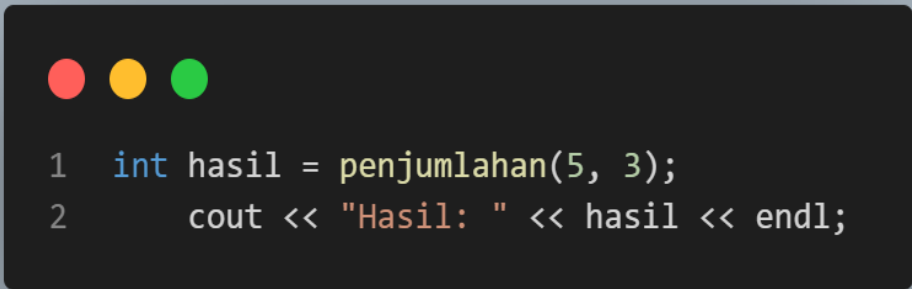
Kode void greet(string name){ cout << "Hello, " << name << "!" << endl; } ini berfungsi untuk membuat sebuah fungsi yang bernama greet.

Fungsi greet ini digunakan untuk menampilkan pesan sapaan di layar.

string name: Artinya, fungsi ini menerima satu input berupa teks (string) yang akan kita sebut sebagai name.

cout << "Hello, " << name << "!" << endl;; Bagian ini akan mencetak kalimat "Hello, [nama]!" ke layar, dengan kata "[nama]" diganti dengan nilai yang kita berikan saat memanggil fungsi ini.

## 3 .Menggunakan fungsi penjumlahan :



```
1 int hasil = penjumlahan(5, 3);
2     cout << "Hasil: " << hasil << endl;
```

Hitung penjumlahan: Bagian penjumlahan(5, 3) memanggil fungsi yang sudah dibuat sebelumnya untuk menjumlahkan angka 5 dan 3.

Simpan hasil: Hasil penjumlahan (yaitu 8) disimpan ke dalam variabel yang bernama hasil.

Tampilkan hasil: Bagian `cout << "Hasil: " << hasil << endl;` akan menampilkan kalimat "Hasil: 8" di layar.

#### 4. Memanggil fungsi greet :



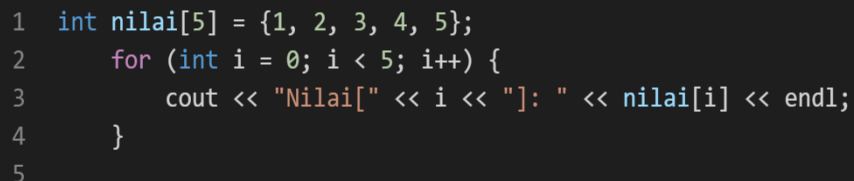
```
1 greet("Haza");
```

Baris kode ini melakukan satu hal utama: memanggil fungsi greet dengan argumen "Haza".

greet: Ini adalah nama fungsi yang telah didefinisikan sebelumnya. Fungsi ini dirancang khusus untuk menampilkan pesan sapaan.

"Haza": Ini adalah argumen yang diberikan kepada fungsi greet. Argumen ini akan menjadi nilai yang akan digunakan oleh fungsi greet untuk mengganti placeholder dalam pesan sapaannya.

#### 5. Menampilkan elemen Array 1 dimensi :



```
1 int nilai[5] = {1, 2, 3, 4, 5};
2     for (int i = 0; i < 5; i++) {
3         cout << "Nilai[" << i << "]: " << nilai[i] << endl;
4     }
5
```

Kode ini bertujuan untuk menampilkan semua elemen yang ada di dalam sebuah array satu dimensi bernama nilai.

```
int nilai[5] = {1, 2, 3, 4, 5};:
```

int nilai[5]: Deklarasi sebuah array dengan nama nilai yang dapat menampung 5 buah bilangan bulat (integer).

= {1, 2, 3, 4, 5}: Inisialisasi array nilai dengan nilai-nilai 1, 2, 3, 4, dan 5. Setiap nilai akan menempati satu indeks dalam array.

```
for (int i = 0; i < 5; i++) { ... }:
```

for: Keyword untuk memulai sebuah perulangan.

int i = 0: Deklarasi variabel i sebagai pengatur perulangan dengan nilai awal 0. Variabel i ini akan digunakan sebagai indeks untuk mengakses elemen-elemen dalam array.

i < 5: Kondisi perulangan. Perulangan akan terus berjalan selama nilai i kurang dari 5.

i++: Setelah setiap iterasi, nilai i akan bertambah 1.

```
cout << "Nilai[" << i << "]: " << nilai[i] << endl;:
```

cout: Objek untuk menampilkan output ke layar.

"Nilai[" << i << "]: ": Menampilkan teks "Nilai[i]: " di mana i akan digantikan dengan nilai indeks saat itu.

nilai[i]: Mengakses elemen array nilai pada indeks ke-i dan menampilkan nilainya.

endl: Memindahkan kursor ke baris baru setelah menampilkan output.

6. Array 2 dimensi :



```
1  int nilai2d[3][4] = {  
2      {1, 2, 3, 4},  
3      {5, 6, 7, 8},  
4      {9, 10, 11, 12}  
5  };
```

Kode ini mendeklarasikan dan menginisialisasi sebuah array dua dimensi bernama nilai2d.

`int nilai2d[3][4]:`

`int`: Menentukan tipe data elemen dalam array, yaitu bilangan bulat (integer).

`nilai2d`: Nama yang diberikan untuk array ini.

`[3][4]`: Menentukan ukuran array, yaitu memiliki 3 baris dan 4 kolom. Artinya, array ini dapat menampung  $3 \times 4 = 12$  elemen.

`{ {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} }:`

Bagian ini digunakan untuk mengisi nilai-nilai awal ke dalam array.

Setiap pasangan kurung kurawal `{}` mewakili satu baris dalam array.

Jadi, baris pertama berisi nilai 1, 2, 3, dan 4, baris kedua berisi 5, 6, 7, 8, dan seterusnya.

7. Loop untuk menampilkan array 2 dimensi :



```
1  for (int i = 0; i < 3; i++) {  
2      for (int j = 0; j < 4; j++) {  
3          cout << nilai2d[i][j] << " ";  
4      }  
5      cout << endl;  
6  }
```

`for (int i = 0; i < 3; i++) { ... }:`

`for`: Keyword untuk memulai sebuah perulangan.

`int i = 0`: Mendeklarasikan variabel `i` sebagai indeks untuk baris, dimulai dari 0.

`i < 3`: Kondisi perulangan, perulangan akan terus berjalan selama `i` kurang dari 3 (jumlah baris).

`i++`: Setelah setiap iterasi baris, nilai `i` akan bertambah 1.

`for (int j = 0; j < 4; j++) { ... }:`

`for`: Keyword untuk memulai perulangan dalam (nested loop).

`int j = 0`: Mendeklarasikan variabel `j` sebagai indeks untuk kolom, dimulai dari 0.

`j < 4`: Kondisi perulangan, perulangan akan terus berjalan selama `j` kurang dari 4 (jumlah kolom).

`j++`: Setelah setiap iterasi kolom, nilai `j` akan bertambah 1.

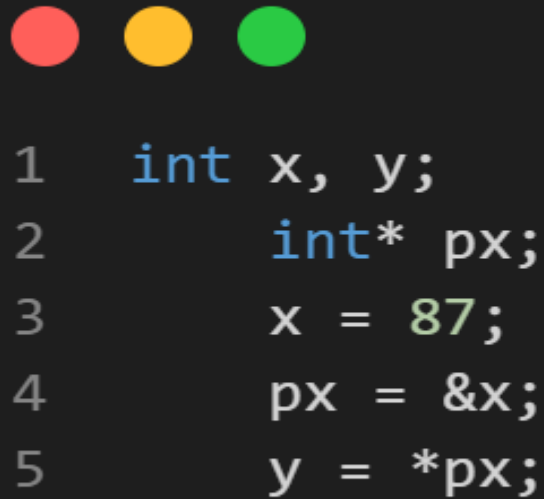
`cout << nilai2d[i][j] << " ";`

Mencetak nilai elemen pada baris ke-`i` dan kolom ke-`j` dari array `nilai2d`, diikuti spasi.

`cout << endl;`

Memindahkan kursor ke baris baru setelah mencetak semua elemen pada satu baris.

8. Pointer :



```
1  int x, y;  
2      int* px;  
3      x = 87;  
4      px = &x;  
5      y = *px;
```

Kode di atas membuat sebuah pointer `px` yang menunjuk ke variabel `x`. Kemudian, nilai dari `x` disalin ke variabel `y` menggunakan pointer `px`.

`int x, y;` Deklarasi dua variabel integer bernama `x` dan `y`.

`int* px;` Deklarasi sebuah pointer bernama `px` yang menunjuk ke variabel integer.


`x = 87;` Memberikan nilai 87 kepada variabel `x`.

`px = &x;` Menugaskan alamat memori dari variabel `x` kepada pointer `px`. Operator `&` digunakan untuk mengambil alamat memori.

`y = *px;` Mengambil nilai yang disimpan pada alamat memori yang ditunjuk oleh `px` dan menyimpannya ke dalam variabel `y`. Operator `*` digunakan untuk dereferensi pointer, yaitu untuk mengambil nilai yang ditunjuk oleh pointer.

9. Menampilkan nilai pointer dan variabel :





```
1  cout << "Alamat x = " << &x << endl;
2      cout << "Isi px = " << px << endl;
3      cout << "Isi x = " << x << endl;
4      cout << "Nilai yang ditunjuk px = " << *px << endl;
5      cout << "Nilai y = " << y << endl;
```

Kode ini digunakan untuk menampilkan berbagai informasi terkait variabel x, y, dan pointer px.

**cout << "Alamat x = " << &x << endl;;**

Mencetak alamat memori dari variabel x. Operator & digunakan untuk mengambil alamat memori dari sebuah variabel.

**cout << "Isi px = " << px << endl;;**

Mencetak nilai dari pointer px. Karena px menyimpan alamat memori, maka yang dicetak adalah alamat memori yang sama dengan alamat x.

**cout << "Isi x = " << x << endl;;**

Mencetak nilai yang tersimpan dalam variabel x.

**cout << "Nilai yang ditunjuk px = " << \*px << endl;;**

Mencetak nilai yang ditunjuk oleh pointer px. Operator \* digunakan untuk dereferensi pointer, artinya kita mengambil nilai yang ada pada alamat memori yang ditunjuk oleh px. Karena px menunjuk ke x, maka nilai yang dicetak sama dengan nilai x.

**cout << "Nilai y = " << y << endl;;**

Mencetak nilai yang tersimpan dalam variabel y. Karena y telah diberikan nilai yang sama dengan nilai yang ditunjuk oleh px, maka nilai y akan sama dengan nilai x.

## V. UNGUIDED



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, angka;
6      cout << "Masukkan jumlah angka: ";
7      cin >> n;
8
9      int arr[n];
10     cout << "Masukkan " << n << " angka: ";
11     for (int i = 0; i < n; i++) {
12         cin >> arr[i];
13     }
14
15     cout << "Data Array: ";
16     for (int i = 0; i < n; i++) {
17         cout << arr[i] << " ";
18     }
19     cout << endl;
20
21     cout << "Nomor Genap: ";
22     for (int i = 0; i < n; i++) {
23         if (arr[i] % 2 == 0) {
24             cout << arr[i] << " ";
25         }
26     }
27     cout << endl;
28
29     cout << "Nomor Ganjil: ";
30     for (int i = 0; i < n; i++) {
31         if (arr[i] % 2 != 0) {
32             cout << arr[i] << " ";
33         }
34     }
35     cout << endl;
36
37     return 0;
38 }
```

```
Masukkan jumlah angka: 10
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 : 31.224 s
Press any key to continue.
```

Deklarasi Variabel:

n: Menyimpan jumlah angka yang akan diinputkan pengguna.

angka: Variabel sementara untuk menyimpan setiap angka yang diinputkan.

arr: Array untuk menyimpan semua angka yang diinputkan.

Input Jumlah Angka:

Meminta pengguna untuk memasukkan jumlah angka yang ingin diproses.

Input Angka:

Menggunakan perulangan for untuk meminta pengguna memasukkan setiap angka dan menyimpannya ke dalam array arr.

Menampilkan Semua Angka:

Menggunakan perulangan for untuk mencetak semua angka yang telah diinputkan.

Memisahkan Angka Genap dan Ganjil:

Menggunakan dua perulangan for yang berbeda

Perulangan pertama untuk mencetak angka genap dengan memeriksa apakah sisa pembagian dengan 2 adalah 0.

Perulangan kedua untuk mencetak angka ganjil dengan memeriksa apakah sisa pembagian dengan 2 tidak sama dengan 0.



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int dimensi1, dimensi2, dimensi3;
6
7      cout << "Masukkan dimensi pertama: ";
8      cin >> dimensi1;
9      cout << "Masukkan dimensi kedua: ";
10     cin >> dimensi2;
11     cout << "Masukkan dimensi ketiga: ";
12     cin >> dimensi3;
13
14     int array3D[dimensi1][dimensi2][dimensi3];
15
16     cout << "Masukkan elemen-elemen array:\n";
17     for (int i = 0; i < dimensi1; i++) {
18         for (int j = 0; j < dimensi2; j++) {
19             for (int k = 0; k < dimensi3; k++) {
20                 cout << "Elemen [" << i << "][" << j << "][" << k << "]: ";
21                 cin >> array3D[i][j][k];
22             }
23         }
24     }
25
26     cout << "\nArray yang telah dibuat:\n";
27     for (int i = 0; i < dimensi1; i++) {
28         for (int j = 0; j < dimensi2; j++) {
29             for (int k = 0; k < dimensi3; k++) {
30                 cout << array3D[i][j][k] << " ";
31             }
32             cout << endl;
33         }
34         cout << endl;
35     }
36
37     return 0;
38 }
39
```

```

Masukkan dimensi pertama: 2
Masukkan dimensi kedua: 3
Masukkan dimensi ketiga: 2
Masukkan elemen-elemen array:
Elemen [0][0][0]: 1
Elemen [0][0][1]: 2
Elemen [0][1][0]: 3
Elemen [0][1][1]: 4
Elemen [0][2][0]: 5
Elemen [0][2][1]: 6
Elemen [1][0][0]: 7
Elemen [1][0][1]: 8
Elemen [1][1][0]: 9
Elemen [1][1][1]: 10
Elemen [1][2][0]: 11
Elemen [1][2][1]: 12

Array yang telah dibuat:
1 2
3 4
5 6

7 8
9 10
11 12

Process returned 0 (0x0)   execution time : 40.906 s
Press any key to continue.
|

```

### Deklarasi Dimensi:

Program meminta pengguna untuk memasukkan tiga nilai integer yang akan menjadi dimensi dari array tiga dimensi.

### Deklarasi Array:

Array tiga dimensi array3D dideklarasikan dengan ukuran yang ditentukan oleh pengguna.

### Input Elemen:

Menggunakan tiga nested loop (perulangan bersarang) untuk mengakses setiap elemen array dan meminta pengguna untuk memasukkan nilai untuk setiap elemen.

### Output Array:

Menggunakan tiga nested loop lagi untuk mencetak semua elemen array yang telah diisi.



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, pilihan;
6      cout << "Masukkan jumlah elemen array: ";
7      cin >> n;
8
9      int arr[n];
10     cout << "Masukkan elemen array:\n";
11     for (int i = 0; i < n; i++) {
12         cin >> arr[i];
13     }
14
15     do {
16         cout << "\nMenu:\n";
17         cout << "1. Cari nilai maksimum\n";
18         cout << "2. Cari nilai minimum\n";
19         cout << "3. Cari nilai rata-rata\n";
20         cout << "4. Keluar\n";
21         cout << "Pilih menu: ";
22         cin >> pilihan;
23
24         switch (pilihan) {
25             case 1: {
26                 int max = arr[0];
27                 for (int i = 1; i < n; i++) {
28                     if (arr[i] > max) {
29                         max = arr[i];
30                     }
31                 }
32                 cout << "Nilai maksimum: " << max << endl;
33                 break;
34             }
35             case 2: {
36                 int min = arr[0];
37                 for (int i = 1; i < n; i++) {
38                     if (arr[i] < min) {
39                         min = arr[i];
40                     }
41                 }
42                 cout << "Nilai minimum: " << min << endl;
43                 break;
44             }
45             case 3: {
46                 int sum = 0;
47                 for (int i = 0; i < n; i++) {
48                     sum += arr[i];
49                 }
50                 double rata_rata = (double)sum / n;
51                 cout << "Nilai rata-rata: " << rata_rata << endl;
52                 break;
53             }
54             case 4:
55                 cout << "Keluar dari program." << endl;
56                 break;
57             default:
58                 cout << "Pilihan tidak valid." << endl;
59         }
60     } while (pilihan != 4);
61
62     return 0;
63 }
```

```
Masukkan jumlah elemen array: 5  
Masukkan elemen array:
```

```
10  
5  
15  
7  
3
```

```
Menu:
```

1. Cari nilai maksimum
2. Cari nilai minimum
3. Cari nilai rata-rata
4. Keluar

```
Pilih menu: 1
```

```
Nilai maksimum: 15
```

Input: Program meminta pengguna untuk memasukkan jumlah elemen array dan nilai-nilai elemennya.

Menu: Program menampilkan menu interaktif dengan pilihan untuk mencari nilai maksimum, minimum, atau rata-rata.

Perulangan: Program akan terus berjalan sampai pengguna memilih opsi untuk keluar (pilihan 4).

Perhitungan:

Nilai Maksimum: Mencari elemen terbesar dalam array.

Nilai Minimum: Mencari elemen terkecil dalam array.

Nilai Rata-rata: Menghitung jumlah semua elemen dan membaginya dengan jumlah elemen.

Output: Program akan menampilkan hasil perhitungan sesuai dengan pilihan pengguna.

## VI. KESIMPULAN

Array, pointer, fungsi, dan prosedur adalah konsep dasar dalam pemrograman C++ yang saling berkaitan, Array digunakan untuk menyimpan data secara terorganisir, Pointer memberikan cara untuk mengakses dan memanipulasi data secara langsung pada tingkat memori, Fungsi dan prosedur digunakan untuk membuat kode lebih modular dan mudah dikelola.