

# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **PERTEMUAN 2**

### **PENGENALAN CPP BAGIAN 2**



**Nama :**

Reyner Atira Prasetyo (2311104057)

S1SE-07-02

**Dosen :**

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

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **I. TUJUAN**

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

## **II. TOOL**

1. Visual Studio Code
2. GCC

## **III. DASAR TEORI**

### **1. Array**

Array merupakan kumpulan data dengan nama yang sama dan setiap elemen bertipe data sama. Untuk mengakses setiap komponen / elemen array berdasarkan indeks dari setiap elemen.

### **2. Array Satu Dimensi**

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

```
tipe_data nama_var[ukuran]
```

Contoh:

```
int nilai[10];
```

Menyatakan bahwa array nilai mengandung 10 elemen dan bertipe integer.

Dalam C++ data array disimpan dalam memori pada lokasi yang berurutan. Elemen pertama memiliki indeks 0 dan elemen selanjutnya memiliki indeks 1 dan seterusnya. Jadi jika terdapat array dengan 5 elemen maka elemen pertama memiliki indeks 0 dan elemen terakhir memiliki indeks 4.

```
nama_var[indeks]
```

nilai[5] → elemen ke-5 dari array nilai. Contoh memasukkan data ke dalam array :

```
nilai[4] = 90;    /*memasukkan 90 ke dalam array nilai indeks ke-4*/
```

```
cin << nilai[4]  /*membaca input-an dari keyboard*/
```

### **3. Array 2 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, inisialisasi, 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

#### 4. Array Berdimensi Banyak

Merupakan array yang mempunyai indeks banyak, lebih dari dua. Indeks inilah yang menyatakan dimensi array. Array berdimensi banyak lebih susah dibayangkan, sejalan dengan jumlah dimensi dalam array.

Cara deklarasi:

```
tipe_data nama_var[ukuran1][ukuran2]...[ukuran-N];
```

Contoh:

```
int data_rumit[4][6][6];
```

Array sebenarnya masih banyak pengembangannya untuk penyimpanan berbagai bentuk data, pengembangan array misalnya untuk array tak berukuran.

#### 5. Pointer

- Data dan Memori

Digambarkan sebuah memory diasumsikan setiap cell menyimpan 1 byte data. Pada saat komputer pertama kali berjalan keadaan memori adalah kosong. Saat variabel dideklarasikan, OS akan mencari cell kosong untuk dialokasikan sebagai memori variabel tersebut.

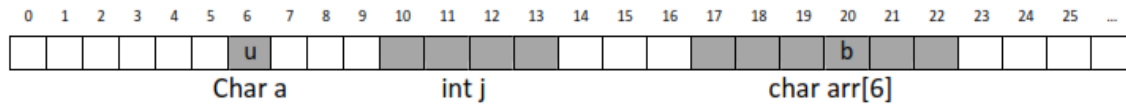
```
char a;
```

```
int j;
```

```
char arr[6];
```

```
arr[3] = 'b'
```

```
a = 'u'
```



Pada contoh di atas variabel `a` dialokasikan di memory alamat `x6`, variabel `j` dialokasikan di alamat `x10-13`, dan variabel `arr` dialokasikan di alamat `x17-22`. Nilai variabel yang ada di dalam memori dapat dipanggil menggunakan alamat dari cell yang menyimpannya. Untuk mengetahui alamat memori tempat di mana suatu variabel dialokasikan, kita bisa menggunakan keyword “&” yang ditempatkan di depan nama variabel yang ingin kita cari alamatnya.

C++	Output	Keterangan
<code>Cout &lt;&lt; a &lt;&lt; endl;</code>	<code>'u'</code>	Nilai variabel <code>a</code>
<code>Cout &lt;&lt; &amp;a &lt;&lt; endl;</code>	<code>x6</code>	Alamat variabel <code>a</code>
<code>Cout &lt;&lt; j &lt;&lt; endl;</code>	<code>0</code>	Nilai variabel <code>j</code>
<code>Cout &lt;&lt; &amp;j &lt;&lt; endl;</code>	<code>x10</code>	Alamat variabel <code>j</code>
<code>Cout &lt;&lt; &amp;(arr[4]) &lt;&lt; endl;</code>	<code>x21</code>	Alamat variabel <code>arr[4]</code>

- **Pointer dan Alamat**

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;
```

Pernyataan di atas berarti bahwa `p_int` diberi nilai berupa alamat dari variabel `j`. Setelah pernyataan tersebut di eksekusi maka dapat dikatakan bahwa `p_int` menunjuk ke variabel `j`. Jika suatu variabel sudah ditunjuk oleh pointer. Maka, variabel yang ditunjuk oleh pointer dapat diakses melalui variabel itu sendiri ataupun melalui pointer.

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

Pointer juga merupakan variabel, karena itu pointer juga akan menggunakan space memory dan memiliki alamat sendiri

```
int *p_int;
```



Gambar 2-4 Ilustrasi Alokasi Pointer

C++	Output	Keterangan
<pre>int j,k; j =10; int *p_int; p_int = &amp;j; cout&lt;&lt; j &lt;&lt; endl; cout&lt;&lt; &amp;j &lt;&lt; endl; cout&lt;&lt; p_int &lt;&lt; endl; cout&lt;&lt; &amp;p_int &lt;&lt; endl; cout&lt;&lt; *p_int &lt;&lt; endl; k = *p_int; cout &lt;&lt; k &lt;&lt; endl;</pre>	<pre>10 X6 X6 X1 10 10</pre>	<pre>Nilai variabel j Alamat variabel j Nilai variabel p_int Alamat variabel p_int Nilai variabel yang ditunjuk p_int Nilai variabel k</pre>

Berikut ini contoh program sederhana menggunakan pointer:

```
1  #include <iostream>
2  #include <conio.h>
3  using namespace std;
4
5  int main(){
6      int x,y; //x dan y bertipe int
7      int *px; //px merupakan variabel pointer menunjuk ke
variabel int
8      x =87;
9      px=&x;
10     y=*px;
11     cout<<"Alamat x= "<<&x<<endl;
12     cout<<"Isi px= "<<px<<endl;
13     cout<<"Isi X= "<<x<<endl;
14     cout<<"Nilai yang ditunjuk px= "<<*px<<endl;
15     cout<<"Nilai y= "<<y<<endl;
16     getch();
17     return 0;
18 }
```

Output :

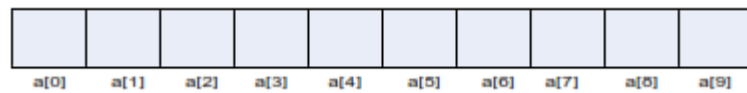
```
Alamat x= 0022FF14
Isi px= 0022FF14
Isi X= 87
Nilai yang ditunjuk px= 87
Nilai y= 87
```

- Pointer dan Array

Ada keterhubungan yang kuat antara array dan pointer. Banyak operasi yang bisa dilakukan dengan array juga bisa dilakukan dengan pointer. Pendeklarasian array: `int a[10];`

Mendefinisikan array sebesar 10, kemudian blok dari objek array tersebut diberi nama

`a[0],a[1],a[2],.....a[9].`



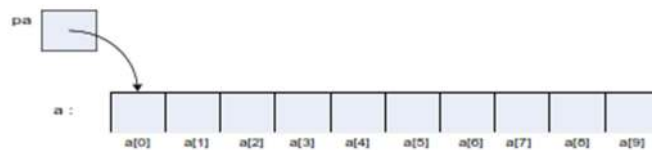
Notasi  $a[i]$  akan merujuk elemen ke- $i$  dari array. Jika  $pa$  merupakan pointer yang menunjuk variabel bertipe integer, yang di deklarasikan sebagai berikut :

```
int *pa;
```

maka pernyataan :

```
pa=&a[0];
```

akan membuat  $pa$  menunjuk ke alamat dari elemen ke-0 dari variabel  $a$ . Sehingga,  $pa$  akan mengandung alamat dari  $a[0]$ .



- Pointer dan String

#### A. String

String merupakan kumpulan karakter yang digunakan untuk memanipulasi data teks. Dalam C++, string dapat direpresentasikan sebagai array karakter atau menggunakan tipe data bawaan `std::string`. Namun, pada pembahasan ini fokus pada penggunaan array karakter. Contoh deklarasi string dalam C++ adalah sebagai berikut:

```
char nama[50];
```

Angka 50 menyatakan kapasitas maksimal karakter yang dapat disimpan dalam array string tersebut.

Untuk memasukkan data string dari keyboard, bisa digunakan `cin`:

```
cin >> nama;
```

Namun, `cin` hanya akan membaca string hingga spasi pertama. Untuk menangkap seluruh input termasuk spasi, dapat digunakan `cin.getline()` atau `std::getline()` dengan `std::string`. Contoh:

```
cin.getline(nama, 50);
```

Inisialisasi string juga dapat dilakukan saat deklarasi. Contoh inisialisasi string dalam bentuk array karakter:

```
char nama[] = "strukdat";
```

Untuk menampilkan string, dapat digunakan `cout`:

```
cout << nama;
```

Akses terhadap karakter dalam string dilakukan menggunakan indeks:

```
cout << nama[3]; // Menampilkan karakter pada indeks ke-3
```

## B. Pointer dan String

String seperti "I am string" dalam C++ tetap merupakan array karakter yang diakhiri dengan karakter null (\0). Untuk mengakses string ini, pointer dapat digunakan. Sebagai contoh:

```
char *pmessage;  
  
pmessage = "now is the time";
```

Pada kode di atas, pmessage adalah pointer yang menunjuk ke awal array karakter string. Pointer tersebut tidak menyalin isi string, hanya menunjuk ke lokasi memori dari string.

Perbedaan mendasar antara array string dan pointer string dalam C++ dapat dilihat dari contoh berikut:

```
char amessage[] = "now is the time"; // Array karakter, isi dapat diubah
```

```
char *pmessage = "now is the time"; // Pointer ke string literal, isi tidak dapat diubah
```

Array amessage merupakan array karakter yang dapat diubah isinya karakter per karakter. Namun, pmessage adalah pointer yang menunjuk ke string literal, di mana isi string tersebut tidak dapat diubah, meskipun pointer dapat diarahkan ke lokasi memori lain.

- 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 ;  
}
```

Jika penentu\_tipe fungsi merupakan tipe dari nilai balik fungsi, bila tidak disebutkan maka akan dianggap (default) sebagai int.

Algoritma	C++
Program coba_fungsi Kamus x,y,z : integer function max3(input: a,b,c : integer) : integer Algoritma input(x,y,z) output( max3(x,y,z) ) function max3(input:a,b,c : integer) : integer kamus temp_max : integer algoritma temp_max ← a if (b>temp_max) then temp_max ← b if (c>temp_max) then temp_max ← c → temp_max	<pre> #include &lt;conio.h&gt; #include &lt;iostream&gt; #include &lt;stdlib.h&gt; using namespace std; int maks3(int a, int b, int c); /*mendeklarasikan prototype fungsi */ int main(){     system("cls");     int x,y,z;     cout&lt;&lt;"masukkan nilai bilangan ke-1 =";     cin&gt;&gt;x;     cout&lt;&lt;"masukkan nilai bilangan ke-2 =";     cin&gt;&gt;y;     cout&lt;&lt;"masukkan nilai bilangan ke-3 =";     cin&gt;&gt;z;     cout&lt;&lt;"nilai maksimumnya adalah ="     &lt;&lt;maks3(x,y,z);     getch();     return 0; } /*badan fungsi */ int maks3(int a, int b, int c){ /* deklarasi variabel lokal dalam fungsi */     Int temp_max =a;     if (b&gt;temp_max)         temp_max=b;     if (c&gt;temp_max)         temp_max=c;     return (temp_max); } </pre>

- Prosedur

Dalam C++, prosedur merupakan fungsi yang tidak mengembalikan nilai, ditandai dengan kata kunci void di depan nama fungsi. Pada dasarnya, C++ tidak membedakan antara fungsi dan prosedur. Setiap prosedur adalah fungsi yang tidak memiliki nilai balik. Prosedur umumnya digunakan untuk menjalankan serangkaian perintah tanpa perlu mengembalikan hasil.

Bentuk umum prosedur dalam C++ adalah sebagai berikut:

```

void nama_prosedur(daftar_parameter) {
    // Blok pernyataan
}

```

- Parameter Fungsi

1. Parameter Formal

Parameter formal adalah variabel yang ada dalam definisi fungsi.  
Contoh:

```
float perkalian(float x, float y) { return (x * y); }
```

2. Parameter Aktual

Parameter aktual adalah nilai atau variabel yang digunakan saat



memanggil fungsi:

X = perkalian(a, b); // a dan b adalah parameter aktual

Y = perkalian(10, 30); // 10 dan 30 adalah parameter actual

## Cara Melewatkan Parameter

### A. Call by Value

Nilai parameter aktual disalin ke parameter formal. Perubahan di dalam fungsi tidak memengaruhi parameter aktual:

```
void fungsi(int x) { x = 10; }
```

### B. Call by Pointer

Alamat variabel dikirimkan ke fungsi, memungkinkan perubahan langsung pada variabel aktual:

```
void tukar(int *px, int *py) {  
    int temp = *px;  
    *px = *py;  
    *py = temp;  
}  
tukar(&a, &b); // Memanggil fungsi dengan alamat a dan b
```

### C. Call by Reference

Alamat variabel dilewatkan secara implisit menggunakan referensi, memungkinkan modifikasi variabel aktual tanpa menggunakan pointer:

```
void tukar(int &px, int &py) {  
    int temp = px;  
    px = py;  
    py = temp;  
}  
tukar(a, b); // Memanggil fungsi dengan referensi
```

## IV. GUIDED

### 1. Array 1 Dimensi



```
1  int main(){  
2  
3      int nilai[5]={1,2,3,4,5};  
4      cout << nilai[0];  
5      cout << nilai[1];  
6      cout << nilai[2];  
7  }
```

Output :

```
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided> &
Launcher.exe' '--stdin=Microsoft-MIEngine-In-gj0vhsy
--pid=Microsoft-MIEngine-Pid-4jyp035v.sua' '--dbgExe
123
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided>
```

Menggunakan Looping :

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

Output :

```
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided> &
--pid=Microsoft-MIEngine-Pid-epeysbfw.s3x' '--dbgExe
1
2
3
4
5
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided>
```

## 2. Array 2 Dimensi

```
1 int nilai[3][4] = {
2     {1,2,3,4},
3     {5,6,7,8},
4     {9,10,11,12}
5 };
6
7 for(int i=0; i<3; i++){
8     for(int j=0; j<4; j++){
9         cout << nilai[i][j] << " ";
10    }
11
12    cout << endl;
13 }
```

Output :

```

PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided> &
Launcher.exe' '--stdin=Microsoft-MIEngine-In-jve5rrb
--pid=Microsoft-MIEngine-Pid-cwp51m1p.tm0' '--dbgExe
1 2 3 4
5 6 7 8
9 10 11 12
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided>

```

### 3. Pointer (Call by Pointer)

```

1  int x,y;
2  int *px;
3  x = 87;
4  px = &x;
5  y= *px;
6
7  cout << "Alamat x = " << &x << endl;
8  cout << "Isi px = " << px << endl;
9  cout << "Isi x= " << x << endl;
10 cout << "Nilai yang ditunjuk px=" << *px << endl;
11 cout << "Nilai y = " << y << endl;
12 getch();

```

Output :

```

--pid=Microsoft-MIEngine-Pid-e3ukm20i.hez' '--dbg
Alamat x = 0x5ffe90
Isi px = 0x5ffe90
Isi x= 87
Nilai yang ditunjuk px=87
Nilai y = 87
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided>

```

### 4. Pointer (Call by Value)

```

1  int penjumlahan (int a, int b){
2      return a + b;
3  }
4
5  int main(){
6      int hasil = penjumlahan(5,3);
7
8      cout << "hasil " << hasil << endl;
9  }

```

Output :

```
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided>
Launcher.exe' '--stdin=Microsoft-MIEngine-In-u0qfr
--pid=Microsoft-MIEngine-Pid-jibzqna5.bga' '--dbgE
hasil 8
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided>
```

## 5. Pointer (Call by Reference)



```
1 void greet(string name){
2     cout << "Hello, " << name << "!" <<
    endl;
3 }
4
5 int main(){
6
7     greet("turbo granny");
8
9 }
```

Output :

```
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided> & '
Launcher.exe' '--stdin=Microsoft-MIEngine-In-opf2stif
--pid=Microsoft-MIEngine-Pid-t2gnjyxx.ea5' '--dbgExe=
Hello, turbo granny!
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\guided>
```

## V. 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  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6
7      //ukuran array
8      cout << "Masukkan jumlah elemen dalam array: ";
9      cin >> n;
10
11     int data_array[n], nomor_genap[n], nomor_ganjil[n];
12     int genap_count = 0, ganjil_count = 0;
13
14     //memasukan data ke dalam array
15     cout << "Masukkan angka ke dalam array: ";
16     for (int i = 0; i < n; i++) {
17         cin >> data_array[i];
18
19         if (data_array[i] % 2 == 0) {
20             nomor_genap[genap_count++] = data_array[i];
21         } else {
22             nomor_ganjil[ganjil_count++] = data_array[i];
23         }
24     }
25
26     //output
27     cout << "Data Array : ";
28     for (int i = 0; i < n; i++) {
29         cout << data_array[i] << " ";
30     }
31     cout << endl;
32
33     cout << "Nomor Genap : ";
34     for (int i = 0; i < genap_count; i++) {
35         cout << nomor_genap[i];
36         if (i < genap_count - 1) {
37             cout << ", ";
38         }
39     }
40     cout << "," << endl;
41
42     cout << "Nomor Ganjil : ";
43     for (int i = 0; i < ganjil_count; i++) {
44         cout << nomor_ganjil[i];
45         if (i < ganjil_count - 1) {
46             cout << ", ";
47         }
48     }
49     cout << "," << endl;
50
51     return 0;
52 }
53

```

Output :

```

PS D:\PRAKTIKUM\Struktur Data\pertemuan2\unguided>
ugLauncher.exe' '--stdin=Microsoft-MIEngine-In-hjm5m
'--pid=Microsoft-MIEngine-Pid-10fbhmp0.3gl' '--dbgE
Masukkan jumlah elemen dalam array: 6
Masukkan angka ke dalam array: 1
2
3
4
5
6
Data Array : 1 2 3 4 5 6
Nomor Genap : 2, 4, 6,
Nomor Ganjil : 1, 3, 5,
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\unguided>

```

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

Source Code :

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x, y, z;
6
7      //ukuran array
8      cout << "Masukkan ukuran dimensi pertama (x): ";
9      cin >> x;
10     cout << "Masukkan ukuran dimensi kedua (y): ";
11     cin >> y;
12     cout << "Masukkan ukuran dimensi ketiga (z): ";
13     cin >> z;
14
15     //membuat array
16     int array3D[x][y][z];
17
18     //memasukan elemen
19     cout << "Masukkan elemen array tiga dimensi:\n";
20     for (int i = 0; i < x; i++) {
21         for (int j = 0; j < y; j++) {
22             for (int k = 0; k < z; k++) {
23                 cout << "Elemen [" << i << "][" << j << "][" << k << "]: ";
24                 cin >> array3D[i][j][k];
25             }
26         }
27     }
28
29     cout << "\nArray 3D yang dimasukkan adalah:\n";
30     for (int i = 0; i < x; i++) {
31         for (int j = 0; j < y; j++) {
32             for (int k = 0; k < z; k++) {
33                 cout << "Elemen [" << i << "][" << j << "][" << k << "] = " << array3D[i][j][k] << endl;
34             }
35         }
36     }
37
38     //output
39     cout << "\nTampilan keseluruhan array 3D:\n";
40     for (int i = 0; i < x; i++) {
41         cout << "Lapisan ke-" << i << ":\n";
42         for (int j = 0; j < y; j++) {
43             for (int k = 0; k < z; k++) {
44                 cout << array3D[i][j][k] << " "; //tampilkan setiap elemen
45             }
46             cout << endl; // ganti baris setiap baris y selesai
47         }
48         cout << endl; // spasi antar x
49     }
50
51     return 0;
52 }
53

```

Output :

```
ugLauncher.exe' '--stdin=Microsoft-MIEngine-In-2ujn6
'--pid=Microsoft-MIEngine-Pid-das05sce.ip5' '--dbg
Masukkan ukuran dimensi pertama (x): 2
Masukkan ukuran dimensi kedua (y): 2
Masukkan ukuran dimensi ketiga (z): 2
Masukkan elemen array tiga dimensi:
Elemen [0][0][0]: 1
Elemen [0][0][1]: 2
Elemen [0][1][0]: 3
Elemen [0][1][1]: 1
Elemen [1][0][0]: 2
Elemen [1][0][1]: 3
Elemen [1][1][0]: 4
Elemen [1][1][1]: 5

Array 3D yang dimasukkan adalah:
Elemen [0][0][0] = 1
Elemen [0][0][1] = 2
Elemen [0][1][0] = 3
Elemen [0][1][1] = 1
Elemen [1][0][0] = 2
Elemen [1][0][1] = 3
Elemen [1][1][0] = 4
Elemen [1][1][1] = 5

Tampilan keseluruhan array 3D:
Lapisan ke-0:
1 2
3 1

Lapisan ke-1:
2 3
4 5

PS D:\PRAKTIKUM\Struktur Data\pertemuan2\unguided>
```

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

Source Code :

```

1  #include <iostream>
2  using namespace std;
3
4  //untuk mencari nilai maksimum
5  int cariMaksimum(int arr[], int n) {
6      int maksimum = arr[0];
7      for (int i = 1; i < n; i++) {
8          if (arr[i] > maksimum) {
9              maksimum = arr[i];
10         }
11     }
12     return maksimum;
13 }
14
15 //untuk mencari nilai minimum
16 int cariMinimum(int arr[], int n) {
17     int minimum = arr[0];
18     for (int i = 1; i < n; i++) {
19         if (arr[i] < minimum) {
20             minimum = arr[i];
21         }
22     }
23     return minimum;
24 }
25
26 //untuk mencari nilai rata-rata
27 double cariRataRata(int arr[], int n) {
28     int jumlah = 0;
29     for (int i = 0; i < n; i++) {
30         jumlah += arr[i];
31     }
32     return (double)jumlah / n;
33 }
34
35 int main() {
36     int n, pilihan;
37
38     //input jumlah elemen array
39     cout << "Masukkan jumlah elemen array: ";
40     cin >> n;
41
42     //input elemen array
43     int arr[n];
44     cout << "Masukkan elemen array:\n";
45     for (int i = 0; i < n; i++) {
46         cin >> arr[i];
47     }
48
49     do {
50         cout << "\nMenu Operasi Array:\n";
51         cout << "1. Cari Nilai Maksimum\n";
52         cout << "2. Cari Nilai Minimum\n";
53         cout << "3. Cari Nilai Rata-Rata\n";
54         cout << "4. Keluar\n";
55         cout << "Masukkan pilihan: ";
56         cin >> pilihan;
57
58         switch (pilihan) {
59             case 1:
60                 cout << "Nilai Maksimum: " << cariMaksimum(arr, n) << endl;
61                 break;
62             case 2:
63                 cout << "Nilai Minimum: " << cariMinimum(arr, n) << endl;
64                 break;
65             case 3:
66                 cout << "Nilai Rata-Rata: " << cariRataRata(arr, n) << endl;
67                 break;
68             case 4:
69                 cout << "Keluar dari program.\n";
70                 break;
71             default:
72                 cout << "Pilihan tidak valid, coba lagi.\n";
73                 break;
74         }
75
76     } while (pilihan != 4);
77
78     return 0;
79 }
80

```



Output :

```
PS D:\PRAKTIKUM\Struktur Data\pertemua
uglauncher.exe' '--stdin=Microsoft
'--pid=Microsoft-MIEngine-Pid-dpz
Masukkan jumlah elemen array: 8
Masukkan elemen array:
2
2
3
4
7
8
1
1
```

```
Menu Operasi Array:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-Rata
4. Keluar
Masukkan pilihan: 1
Nilai Maksimum: 8
```

```
Menu Operasi Array:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-Rata
4. Keluar
Masukkan pilihan: 2
Nilai Minimum: 1
```

```
Menu Operasi Array:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-Rata
4. Keluar
Masukkan pilihan: 3
Nilai Rata-Rata: 3.5
```

```
Menu Operasi Array:
1. Cari Nilai Maksimum
2. Cari Nilai Minimum
3. Cari Nilai Rata-Rata
4. Keluar
Masukkan pilihan: 4
Keluar dari program.
```

```
PS D:\PRAKTIKUM\Struktur Data\pertemuan2\unguided> |
```

## **VI. KESIMPULAN**

Dalam praktikum ini, saya telah mempelajari dan berhasil mengimplementasikan konsep array dan pointer dalam bahasa pemrograman C++. Penggunaan array memungkinkan penyimpanan dan pengelolaan kumpulan data secara efisien, sementara pointer memberikan kemampuan untuk mengakses dan memanipulasi alamat memori secara langsung.

Melalui contoh implementasi, saya memahami bagaimana array dapat digunakan untuk menyimpan data bertipe sama, dan bagaimana pointer dapat menunjuk ke elemen-elemen dalam array. Saya juga belajar tentang perbedaan antara pemanggilan dengan nilai, referensi, dan pointer, serta bagaimana cara penggunaan masing-masing metode dapat mempengaruhi performa dan efektivitas program.

Keseluruhan pengalaman ini memperkuat pemahaman saya tentang manajemen memori dan efisiensi dalam pemrograman C++, yang sangat penting dalam pengembangan perangkat lunak. Praktikum ini memberikan dasar yang kuat untuk eksplorasi lebih lanjut tentang struktur data dan teknik pemrograman yang lebih kompleks.