

**LAPORAN PRAKTIKUM**  
**MODUL 2**  
**PENGENALAN BAHASA C++ (BAGIAN KEDUA)**



**Disusun Oleh:**  
**Tiurma Grace Angelina (2311104042)**  
**SE-07-02**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**PURWOKERTO**  
**2024**

## 1. Tujuan

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

## 2. Landasan Teori

Berikut dasar teori untuk memahami penggunaan pointer, alamat memori, serta implementasi fungsi dan prosedur dalam program berdasarkan sumber yang diberikan:

- **Pointer dan Alamat Memori**

Pointer merupakan variabel yang menyimpan alamat memori dari variabel lain, sehingga memungkinkan program untuk mengakses atau memanipulasi data secara tidak langsung. Memori komputer dapat digambarkan sebagai array satu dimensi yang sangat besar di mana setiap sel memiliki alamat unik. Setiap variabel yang dibuat dalam program dialokasikan oleh sistem operasi pada alamat memori tertentu. Dengan menggunakan pointer, kita dapat menyimpan alamat tersebut dan mengakses nilai variabel melalui alamatnya.

Cara kerja pointer dimulai dengan deklarasi seperti berikut:

```
int p_int;
```

Variabel `p_int` merupakan pointer yang menunjuk ke variabel tipe data `int`. Agar pointer menunjuk ke variabel tertentu, ia harus diisi dengan alamat variabel tersebut, misalnya:

```
p_int = &j;
```

Dengan ini, `p_int` menunjuk ke alamat dari variabel `j`, dan nilai `j` dapat diakses melalui `p_int`.

- **Implementasi Fungsi**

Fungsi adalah blok kode yang dirancang untuk melaksanakan tugas tertentu. Fungsi memberikan struktur pada program dengan membaginya menjadi modul-modul kecil yang mudah dipahami dan dikembangkan. Umumnya, fungsi memerlukan masukan berupa parameter yang diolah dan kemudian mengembalikan hasil sebagai nilai balik fungsi.

Contoh bentuk umum fungsi:

```
int max3(int a, int b, int c) {  
    int temp_max = a;  
    if (b > temp_max) temp_max = b;  
    if (c > temp_max) temp_max = c;  
    return temp_max;  
}
```

Fungsi ini menerima tiga parameter `a`, `b`, dan `c`, dan mengembalikan nilai terbesar di

### 3. Guided

- Array

1. CODE :

```
guided2.cpp > ...
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  int main()
7  {
8
9      int nilai[5]={1,2,3,4,5};
10     cout << nilai[0];
11     cout << nilai[1];
12     cout << nilai[2];
13     cout << nilai[3];
14     cout << nilai[4];
15
16 }
```

OUTPUT :

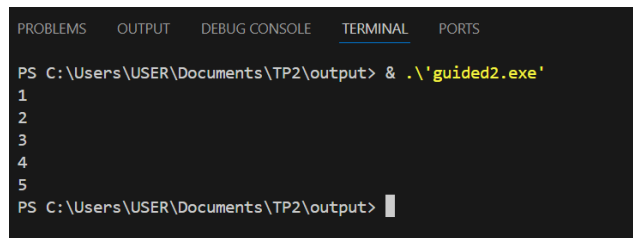
```
PS C:\Users\USER\Documents\TP2\output> cd 'c:\Users\USER\Documents\TP2\output'
PS C:\Users\USER\Documents\TP2\output> & .\guided1.exe
12345
```

- **iostream** adalah pustaka yang digunakan untuk operasi input-output, seperti cout (untuk menampilkan output) dan cin (untuk menerima input dari pengguna).
- **conio.h** digunakan untuk fungsi tambahan seperti getch(), meskipun dalam kode ini getch() tidak digunakan.
- Mendeklarasikan array bernama **nilai** dengan ukuran 5. Array ini berisi lima elemen dengan nilai-nilai 1, 2, 3, 4, 5.
- Indeks array dimulai dari 0, sehingga elemen **nilai[0]** adalah 1, elemen **nilai[1]** adalah 2, dan seterusnya hingga **nilai[4]** yang berisi 5.
- Setiap elemen array ditampilkan menggunakan **cout**. Misalnya: `cout << nilai[0];` Ini menampilkan elemen pertama dari array, yaitu 1.
- Ini dilakukan untuk semua elemen array, dari **nilai[0]** hingga **nilai[4]**, sehingga semua elemen array ditampilkan dalam urutan yang sama (1, 2, 3, 4, 5) tanpa pemisah atau baris baru.

2. CODE :

```
guided1.cpp > ...
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  int main()
7  {
8
9      int nilai[5]={1,2,3,4,5};
10
11     for(int i=0; i<5; i++){
12         cout << nilai[i] << endl;
13     }
14
15 }
```

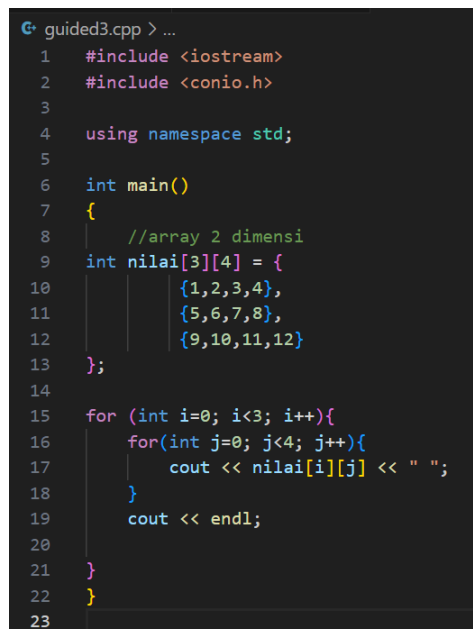
## OUTPUT :



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\USER\Documents\TP2\output> & .\'guided2.exe\'
1
2
3
4
5
PS C:\Users\USER\Documents\TP2\output>
```

- `'iostream'` digunakan untuk operasi input/output, seperti `'cout'` untuk menampilkan teks ke layar.
- `'conio.h'` adalah pustaka yang biasanya digunakan untuk fungsi tambahan seperti `'getch()'` atau `'clrscr()'`, meskipun dalam kode ini tidak digunakan secara eksplisit.
- Mendeklarasikan array bernama `'nilai'` dengan 5 elemen bertipe `'int'`, yang berisi nilai-nilai 1, 2, 3, 4, dan 5.
- Indeks array dimulai dari 0, sehingga `'nilai[0]'` adalah `'1'`, `'nilai[1]'` adalah `'2'`, dan seterusnya hingga `'nilai[4]'` yang berisi `'5'`.
- `'for (int i = 0; i < 5; i++)'`: Loop ini digunakan untuk mengulang sebanyak 5 kali, dari indeks `'0'` hingga `'4'` (karena `**'i < 5'*`).
- Pada setiap iterasi, `'nilai[i]'` menampilkan elemen array pada posisi `i`. Sebagai contoh, saat `'i = 0'`, `'nilai[0]'` menampilkan `'1'`, dan saat `'i = 4'`, `'nilai[4]'` menampilkan `'5'`.
- `'endl'` digunakan untuk menambahkan baris baru setelah setiap elemen ditampilkan, sehingga setiap nilai ditampilkan di baris yang berbeda.

## 3. CODE :



```
guided3.cpp > ...
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  int main()
7  {
8      //array 2 dimensi
9      int nilai[3][4] = {
10         {1,2,3,4},
11         {5,6,7,8},
12         {9,10,11,12}
13     };
14
15     for (int i=0; i<3; i++){
16         for(int j=0; j<4; j++){
17             cout << nilai[i][j] << " ";
18         }
19         cout << endl;
20     }
21 }
22
23
```

## OUTPUT :

```

PS C:\Users\USER\Documents\TP2\output> cd 'c:\Users\USER\Documents\TP2\output'
PS C:\Users\USER\Documents\TP2\output> & .\'guided3.exe'
1 2 3 4
5 6 7 8
9 10 11 12
PS C:\Users\USER\Documents\TP2\output>

```

- Variabel integer **i** dan **j** digunakan sebagai index untuk loop.
- **for** loop luar (**i**) akan berjalan sebanyak 3 iterasi, dari **i=0** hingga **i<3**.
- Pada awal setiap iterasi loop luar, program akan mencetak sebuah newline (**\n**) untuk membuat baris baru.
- **for** loop dalam (**j**) akan berjalan sebanyak **i+1** iterasi, dari **j=0** hingga **j<=i**.
- Pada setiap iterasi loop dalam, program akan mencetak angka **j+1**.
- Setelah loop dalam selesai, program akan mencetak sebuah newline (**\n**) lagi untuk membuat baris baru.
- Setelah loop luar selesai, program akan mencetak sebuah newline (**\n**) terakhir untuk membuat baris baru.

#### 4. CODE :

```

guided4.cpp > ...
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  int main()
7  {
8
9  //pointer
10 int x,y;
11 int *px;
12 x = 87;
13 px = &x;
14 y= *px;
15
16 cout << "alamat x= " << &x << endl;
17 cout << "isi px= " << px << endl;
18 cout << "isi x= " << x << endl;
19 cout << "nilai yang ditunjuk px= " << *px << endl;
20 cout << "nilai y= " << y << endl;
21 getch();
22
23 }

```

#### OUTPUT :

```

PS C:\Users\USER\Documents\TP2\output> cd 'c:\Users\USER\Documents\TP2\output'
PS C:\Users\USER\Documents\TP2\output> & .\'guided4.exe'
alamat x= 0x61ff04
isi px= 0x61ff04
isi x= 87
nilai yang ditunjuk px= 87
nilai y= 87

```

#### Deklarasi Pointer:

- `int *px;` mendeklarasikan pointer `px` yang dapat menyimpan alamat dari variabel bertipe `int`.

### Pemberian Nilai:

- `x = 87;` memberi nilai 87 kepada variabel `x`.
- `px = &x;` menyimpan alamat memori dari `x` ke dalam pointer `px`. Jadi, `px` sekarang "menunjuk" ke `x`.
- `y = *px;` mengambil nilai yang ditunjuk oleh `px` (yaitu nilai `x`, yaitu 87) dan menyimpannya ke dalam variabel `y`.

### Output:

- `cout << "alamat x = " << &x << endl;` menampilkan alamat memori di mana variabel `x` disimpan.
- `cout << "isi px= " << px << endl;` menampilkan nilai dari `px`, yang merupakan alamat memori dari `x`.
- `cout << "isi x= " << x << endl;` menampilkan nilai dari `x`, yaitu 87.
- `cout << "nilai yang ditunjuk px= " << *px << endl;` menampilkan nilai yang ditunjuk oleh `px`, yaitu nilai `x` (87).
- `cout << "nilai y= " << y << endl;` menampilkan nilai dari `y`, yang juga 87.

### 5. CODE:

```
guided5.cpp > ...
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  void greet (string name){
7      cout << " hello " << name << " ! " << endl;
8  }
9  int main( ){
10     greet( "alice" );
11     return 0;
12 }
13
```

### OUTPUT:

```
PS C:\Users\USER\Documents\TP2\output> & .\'guided5.exe'
hello alice !
PS C:\Users\USER\Documents\TP2\output>
```

### `#include <iostream>` dan `#include <conio.h>`:

- `#include <iostream>` digunakan untuk mengimpor library yang berfungsi untuk input dan output di C++, seperti `cout`.
- `#include <conio.h>` (hanya tersedia di beberapa kompiler lama atau spesifik) berfungsi untuk operasi input/output konsol seperti `getch()`, tetapi tidak digunakan dalam kode ini secara langsung.

### `using namespace std;`:

- Ini digunakan agar kita tidak perlu menulis `std::` sebelum fungsi standar seperti `cout`, `cin`, dan lainnya.

### Fungsi greet(string name):

- Fungsi ini memiliki satu parameter bertipe string yang bernama name.
- Di dalam fungsi, perintah cout digunakan untuk mencetak pesan ke layar. Fungsi ini akan menampilkan teks " hello " diikuti dengan nilai name (nama yang dilewatkan ke fungsi) dan tanda seru "!".
- Fungsi endl digunakan untuk membuat baris baru setelah mencetak pesan.

### Fungsi main():

- Fungsi main adalah titik awal dari eksekusi program.
- Pada baris greet("alice");, fungsi greet dipanggil dengan argumen "alice", sehingga program akan mencetak pesan " hello alice !".
- return 0; mengindikasikan bahwa program selesai dijalankan dengan sukses.

## 6. CODE:

```
guided6.cpp > ...
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  int penjumlahan(int a, int b){
7      return a + b;
8  }
9  int main( ){
10     int hasil = penjumlahan (5, 3);
11     std::cout << "hasil " << hasil << std::endl;
12     return 0;
13 }
14
```

## OUTPUT:

```
PS C:\Users\USER\Documents\TP2\output> & .\guided6.exe
hasil 8
PS C:\Users\USER\Documents\TP2\output>
```

Compiled successfully!

### #include <iostream> dan #include <conio.h>:

- #include <iostream> digunakan untuk menggunakan fungsi input/output seperti cout dan cin.
- #include <conio.h> adalah library tambahan yang digunakan dalam beberapa kompiler lama untuk menangani fungsi konsol, tapi tidak diperlukan di program ini.

### using namespace std;:

- Ini memungkinkan kita untuk menggunakan elemen dari namespace std tanpa harus menulis std:: setiap kali kita menggunakan elemen seperti cout.

### Fungsi penjumlahan(int a, int b):

- Fungsi ini menerima dua parameter bertipe integer (a dan b).
- Fungsi ini mengembalikan hasil dari penjumlahan dua bilangan tersebut (a + b).

### Fungsi main():

- Pada baris `int hasil = penjumlahan(5, 3);`, fungsi penjumlahan dipanggil dengan dua argumen, yaitu 5 dan 3. Hasil dari penjumlahan ini (8) disimpan dalam variabel `hasil`.
- `std::cout << "hasil " << hasil << std::endl;` digunakan untuk mencetak teks "hasil " diikuti dengan nilai dari variabel `hasil`, yaitu 8. `std::endl` digunakan untuk menambahkan baris baru setelah output.
- `return 0;` menandakan bahwa program berakhir dengan sukses.

## 4. Unguided

1. Buatlah program yang menerima *input*-an dua buah bilangan betipe float, kemudian memberikan *output*-an hasil penjumlahan, pengurangan, perkalian, dan pembagian dari dua bilangan tersebut.

CODE :

```
unguided1.cpp > main()
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  int main() {
7      int n, input;
8      int dataArray[100], evenNumbers[100], oddNumbers[100];
9      int evenCount = 0, oddCount = 0;
10
11
12      cout << "Masukkan jumlah data: ";
13      cin >> n;
14
15      cout << "Masukkan " << n << " angka:\n";
16      for (int i = 0; i < n; i++) {
17          cin >> input;
18          dataArray[i] = input;
19
20          if (input % 2 == 0) {
21              evenNumbers[evenCount++] = input;
22          } else {
23              oddNumbers[oddCount++] = input;
24          }
25      }
26
27      cout << "Data Array   : ";
28      for (int i = 0; i < n; i++) {
29          cout << dataArray[i] << " ";
30      }
```



```

unguided1.cpp > main()
6   int main() {
27   cout << "Data Array  : ";
28   for (int i = 0; i < n; i++) {
29       cout << dataArray[i] << " ";
30   }
31   cout << endl;
32
33   cout << "Nomor Genap  : ";
34   for (int i = 0; i < evenCount; i++) {
35       cout << evenNumbers[i];
36       if (i != evenCount - 1) {
37           cout << ", ";
38       }
39   }
40   cout << endl;
41
42   cout << "Nomor Ganjil : ";
43   for (int i = 0; i < oddCount; i++) {
44       cout << oddNumbers[i];
45       if (i != oddCount - 1) {
46           cout << ", ";
47       }
48   }
49   cout << endl;
50
51   getch();
52
53   return 0;
54 }

```

## OUTPUT :

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\USER\Downloads\CPP2VSCODE> cd 'c:\Users\USER\Downloads\CPP2VSCODE\output'
PS C:\Users\USER\Downloads\CPP2VSCODE\output> & .\unguided1.exe
Masukkan jumlah data: 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

```

- include <iostream> digunakan untuk mengimpor library input/output yang memungkinkan penggunaan fungsi seperti cin dan cout.
- include <conio.h> adalah library tambahan untuk fungsi getch(), yang berfungsi untuk menunggu input karakter dari pengguna.
- n: Menyimpan jumlah data yang akan dimasukkan pengguna.
- input: Variabel sementara untuk menyimpan setiap input dari pengguna sebelum dimasukkan ke array.

- `dataArray[100]`: Array untuk menyimpan semua angka yang dimasukkan pengguna.
- `evenNumbers[100]`: Array untuk menyimpan angka-angka genap.
- `oddNumbers[100]`: Array untuk menyimpan angka-angka ganjil.
- `evenCount` dan `oddCount`: Variabel penghitung yang digunakan untuk menghitung jumlah angka genap dan ganjil.
- Program meminta pengguna untuk memasukkan jumlah angka yang akan dimasukkan (`n`).
- Setelah itu, pengguna diminta memasukkan `n` angka satu per satu.
- Setiap angka disimpan di array `dataArray`.
- Program memeriksa apakah angka tersebut genap atau ganjil:
- Genap: Jika angka dibagi 2 memiliki sisa 0 (`input % 2 == 0`), angka tersebut dimasukkan ke array `evenNumbers` dan penghitung `evenCount` bertambah.
- Ganjil: Jika tidak, angka tersebut dimasukkan ke array `oddNumbers` dan penghitung `oddCount` bertambah.
- Program mencetak semua angka yang dimasukkan pengguna (data dari `dataArray`), dipisahkan oleh spasi.
- Program mencetak semua angka genap yang telah disimpan dalam array `evenNumbers`, dipisahkan oleh koma. Jika angka tersebut bukan yang terakhir dalam daftar, koma ditambahkan setelah angka.
- Program mencetak semua angka ganjil yang telah disimpan dalam array `oddNumbers`, juga dipisahkan oleh koma. Sama seperti pada angka genap, koma hanya ditambahkan jika angka tersebut bukan angka terakhir.
- `getch()` digunakan untuk menunggu input pengguna sebelum program berakhir. Ini berguna agar jendela konsol tidak langsung menutup setelah output ditampilkan (berlaku pada beberapa compiler lama).

2. Buatlah sebuah program yang menerima masukan angka dan mengeluarkan *output* nilai angka tersebut dalam bentuk tulisan. Angka yang akan di- *input*-kan user adalah bilangan bulat positif mulai dari 0 s.d 100. CODE :

```

G: unguided2.cpp > ...
1  #include <iostream>
2  #include <conio.h>
3  using namespace std;
4
5  int main() {
6      int x, y, z;
7
8      cout << "Masukkan ukuran dimensi pertama: ";
9      cin >> x;
10     cout << "Masukkan ukuran dimensi kedua: ";
11     cin >> y;
12     cout << "Masukkan ukuran dimensi ketiga: ";
13     cin >> z;
14
15     int array[x][y][z];
16
17     cout << "Masukkan elemen array:\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 << "Array[" << i << "][" << j << "][" << k << "]: ";
22                 cin >> array[i][j][k];
23             }
24         }
25     }
26
27     cout << "\nIsi array 3 dimensi:\n";
28     for (int i = 0; i < x; i++) {
29         for (int j = 0; j < y; j++) {
30             for (int k = 0; k < z; k++) {
31                 cout << array[i][j][k] << " ";
32             }
33             cout << "\n";
34         }
35     }
36 }

```

## OUTPUT :

```

PS C:\Users\USER\Documents\TP2\output> & .\'unguided2.exe'
Masukkan ukuran dimensi pertama: 2
Masukkan ukuran dimensi kedua: 2
Masukkan ukuran dimensi ketiga: 2
Masukkan elemen array:
Array[0][0][0]: 1
Array[0][0][1]: 2
Array[0][1][0]: 3
Array[0][1][1]: 4
Array[1][0][0]: 5
Array[1][0][1]: 6

```

```

Array[1][0][0]: 5
Array[1][0][1]: 6
Array[1][1][0]: 7
Array[1][1][1]: 8

```

```

Isi array 3 dimensi:
1 2
3 4

5 6
7 8

```

- `#include <iostream>`: Digunakan untuk mengimpor library input/output sehingga memungkinkan penggunaan fungsi `cin` untuk input dan `cout` untuk output.
- `#include <conio.h>`: Digunakan untuk menggunakan fungsi `getch()` yang menunggu input karakter sebelum program berakhir (berlaku pada beberapa compiler tertentu).

- **x, y, z:** Variabel yang digunakan untuk menyimpan ukuran dimensi pertama, kedua, dan ketiga dari array tiga dimensi yang akan dibuat.
- Program meminta pengguna untuk memasukkan ukuran dari setiap dimensi (pertama, kedua, dan ketiga) untuk array tiga dimensi. Ukuran ini akan digunakan untuk membuat array dinamis dengan dimensi yang ditentukan oleh input pengguna.
- Program mendeklarasikan array tiga dimensi dengan ukuran yang telah dimasukkan oleh pengguna. Misalnya, jika pengguna memasukkan  $x=2$ ,  $y=3$ , dan  $z=4$ , maka array akan berukuran  $2 \times 3 \times 4$ .
- Program meminta pengguna untuk mengisi elemen-elemen dari array tiga dimensi.
- Pengulangan for bertingkat (3 kali) digunakan untuk mengakses setiap elemen array berdasarkan tiga dimensi ( $x, y, z$ ).
- Setiap elemen array diisi dengan input yang diberikan pengguna.
- Setelah semua elemen diinput, program akan mencetak isi array tiga dimensi tersebut.
- Sama seperti saat menginput, program menggunakan pengulangan for bertingkat untuk mengakses setiap elemen dan menampilkannya.
- Setiap baris elemen dari dimensi kedua dipisahkan dengan baris baru (`endl`), dan setelah setiap lapisan dimensi pertama, ditambahkan baris kosong untuk memisahkannya secara visual.
- `getch()` digunakan untuk menunggu pengguna menekan tombol sebelum menutup program. Ini berguna agar jendela konsol tidak langsung tertutup setelah output ditampilkan, terutama pada beberapa compiler yang menggunakan library `conio.h`.

3. Buatlah program yang dapat memberikan *input* dan *output* sbb:  
CODE :

```
1 #include <iostream>
2 #include <conio.h>
3 using namespace std;
4
5 int main() {
6     int n, pilihan, arr[100], maks, min, total = 0;
7     double rata;
8
9     cout << "Masukkan jumlah elemen array: ";
10    cin >> n;
11    cout << "Masukkan elemen array:\n";
12    for (int i = 0; i < n; i++) {
13        cin >> arr[i];
14        if (i == 0) {
15            maks = min = arr[i];
16        }
17        if (arr[i] > maks) maks = arr[i];
18        if (arr[i] < min) min = arr[i];
19        total += arr[i];
20    }
21    rata = (double)total / n;
22
23    do {
24        cout << "\nMenu:\n1. Maksimum\n2. Minimum\n3. Rata-rata\n4. Keluar\nPilih: ";
25        cin >> pilihan;
26        switch (pilihan) {
27            case 1: cout << "Nilai Maksimum: " << maks << endl; break;
28            case 2: cout << "Nilai Minimum: " << min << endl; break;
29            case 3: cout << "Nilai Rata-rata: " << rata << endl; break;
30        }
31        getch();
32    } while (pilihan != 4);
33
34    return 0;
35 }
36
37
```

## OUTPUT :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\USER\Downloads\CPP2VSCODE\output> cd 'c:\Users\USER\Downloads\CPP2VSCODE\output'
PS C:\Users\USER\Downloads\CPP2VSCODE\output> & .\unguided32.exe
Masukkan jumlah elemen array: 3
Masukkan elemen array:
1
2
3

Menu:

1. Maksimum
2. Minimum
3. Rata-rata
4. Keluar
Pilih: 1
Nilai Maksimum: 3

Menu:
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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

Menu:
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

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

Menu:
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Menu:
1. Maksimum
2. Minimum
3. Rata-rata
4. Keluar
Pilih: 4
```

- **#include <iostream>**: Digunakan untuk input/output seperti cin dan cout.
- **#include <conio.h>**: Digunakan untuk fungsi getch(), yang menunggu input karakter sebelum melanjutkan (berguna agar tampilan menu tidak langsung hilang).
- **n**: Menyimpan jumlah elemen array yang dimasukkan pengguna.
- **pilihan**: Menyimpan pilihan menu dari pengguna.
- **arr[100]**: Array yang menyimpan hingga 100 elemen yang dimasukkan pengguna.
  - **maks**: Variabel untuk menyimpan nilai maksimum dari array.
  - **min**: Variabel untuk menyimpan nilai minimum dari array.
  - **total**: Menyimpan jumlah total semua elemen array (akan digunakan untuk menghitung rata-rata).
  - **rata**: Menyimpan nilai rata-rata dari elemen-elemen array.

- **cin >> n;** Meminta pengguna untuk memasukkan jumlah elemen array.
- Program kemudian meminta pengguna untuk menginput elemen-elemen array satu per satu dan menyimpannya di array `arr[]`.
- **Menentukan nilai maksimum dan minimum:**
- Pada elemen pertama, nilai tersebut akan langsung dijadikan nilai awal untuk variabel maks dan min (`maks = min = arr[i]`).
- Setiap elemen berikutnya akan dibandingkan dengan nilai maksimum dan minimum yang ada:
  - Jika elemen lebih besar dari maks, maka nilai elemen tersebut dijadikan nilai maksimum.
  - Jika elemen lebih kecil dari min, maka nilai elemen tersebut dijadikan nilai minimum.
- Setiap elemen ditambahkan ke variabel total untuk menghitung jumlah total dari seluruh elemen.
- Setelah perulangan selesai, program menghitung rata-rata dengan membagi total jumlah elemen dengan jumlah elemen `n`.
- Total diubah menjadi double agar hasil pembagian tidak hanya berupa bilangan bulat.
- **Menu pilihan:** Setelah data array dimasukkan dan diproses, program menampilkan menu kepada pengguna:
  - **1:** Menampilkan nilai maksimum.
  - **2:** Menampilkan nilai minimum.
  - **3:** Menampilkan nilai rata-rata.
  - **4:** Keluar dari program.
- **cin >> pilihan;** Program membaca input dari pengguna dan mengeksekusi pilihan yang sesuai.
  - Menggunakan **switch** untuk memproses pilihan:
    - Jika pengguna memilih **1**, program menampilkan nilai maksimum.
    - Jika pengguna memilih **2**, program menampilkan nilai minimum.
    - Jika pengguna memilih **3**, program menampilkan nilai rata-rata.
- Setelah setiap pilihan dieksekusi, fungsi `getch()` digunakan untuk menunggu pengguna menekan tombol sebelum menu ditampilkan lagi.
- Perulangan **do-while** memastikan program terus menampilkan menu hingga pengguna memilih pilihan **4** (keluar).
- **getch()** digunakan untuk menunggu pengguna menekan tombol apa pun sebelum melanjutkan ke langkah berikutnya. Fungsinya terutama untuk menjaga agar output dapat dibaca sebelum tampilan berikutnya muncul.





