

LAPORAN PRAKTIKUM
MODUL 2
Pengenalan Bahasa C++ Bagian Kedua



Nama :

Ryan Gabriel Togar Simamora (2311104045)

Kelas : SE07-02

Dosen :

Wahyu Andi Saputra

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. Tujuan

1. Memahami konsep dan penggunaan array untuk menyimpan dan mengakses sekumpulan data dengan tipe yang sama secara efisien.
2. Memahami penggunaan pointer dan alamat memori dalam program C++ untuk mengakses dan memodifikasi data secara langsung melalui alamat memori.
3. Mengimplementasikan fungsi
4. Mengaplikasikan prosedur (fungsi void) untuk menjalankan serangkaian perintah tertentu tanpa mengembalikan nilai, sehingga program dapat melakukan tugas spesifik secara terstruktur.

II. Landasan Teori

A. Array

Array adalah struktur data yang menyimpan sekelompok elemen dengan tipe data yang sama dalam satu variabel, diakses menggunakan indeks. Array memungkinkan penyimpanan dan pengelolaan data secara terstruktur, di mana elemen-elemennya dapat diakses dengan mudah berdasarkan posisi indeksnya.

Indeks array selalu dimulai dari nol 0.

B. Pointer

Dalam bahasa pemrograman C++, pointer adalah jenis variabel khusus yang menyimpan alamat memori dari variabel lain sebagai nilai yang dimilikinya, bukan nilai data langsung itu sendiri. Ini berarti bahwa alih-alih menyimpan data seperti angka atau karakter, pointer menyimpan lokasi atau alamat di mana data tersebut berada dalam memori komputer.

Kemampuan ini memungkinkan pemrogram untuk mengelola memori dengan lebih fleksibel, memberikan akses langsung ke lokasi memori tertentu, serta memanipulasi data yang disimpan di berbagai bagian dari memori program secara efisien. Dengan menggunakan pointer, pemrogram dapat melakukan berbagai operasi tingkat rendah seperti alokasi dan dealokasi memori secara dinamis, serta mengakses dan memodifikasi data melalui alamatnya, yang membuka peluang untuk pengembangan aplikasi yang lebih kompleks dan optimal.

C. Fungsi dan Prosedur

1. Apa itu Fungsi?

Fungsi adalah sub-program yang bisa digunakan kembali baik di dalam program itu sendiri, maupun di program yang lain.

Fungsi dapat menerima input dan menghasilkan output.

Contoh fungsi yang sering kita buat adalah fungsi main().

Fungsi ini memang wajib ada di setiap program C++, karena fungsi inilah yang akan dieksekusi pertama kali saat program berjalan.

Seperti yang saya bilang tadi: "Kalau kita menumpuk semua kode program di dalam fungsi main(), maka program akan menjadi sulit terbaca."

Karena itu, kita perlu membuat fungsi tersendiri agar program tidak berantakan.

Bentuk umum Fungsi adalah sebagai berikut:

```
TipeData NamaFungsi (DaftarParameter){  
    /*Code atau Badan Fungsi*/  
    return nilaireturn;  
}
```

2. Apa itu Prosedur ?

Prosedur merupakan suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian). Prosedur biasanya bersifat suatu aktifitas seperti mencari bilangan prima dari sekumpulan bilangan atau mencari bilangan genap dari sekumpulan bilangan ,dsb. Prosedur biasanya digunakan pada program yang terstruktur karena:

- Merupakan penerapan konsep program modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk prosedur-prosedur.
- Untuk hal-hal yang sering dilakukan / dipakai berulang-ulang, cukup dituliskan sekali saja dalam bentuk prosedur dan dapat dipergunakan atau dipanggil berulang kali jika diperlukan.
- Membuat kode program lebih mudah dibaca / dimengerti terutama oleh programmer lain.
- Dapat digunakan untuk menyembunyikan detail program

Bentuk umum prosedur adalah sebagai berikut:

```
Void NamaProsedur (DaftarParameter){  
    /*Code atau Badan Prosedur*/  
}
```

3. Perbedaan fungsi dengan prosedur

Pada fungsi, tipe data nilai yang dikirimkan balik (return value) terdapat pada nama fungsinya, sedangkan prosedur tidak memiliki return value (misal return luas). Karena nilai balik berada di nama fungsi tersebut, maka fungsi tersebut dapat langsung digunakan untuk dicetak hasilnya. Atau nilai fungsi tersebut dapat juga langsung dipindahkan ke pengenalan variable yang lainnya (misal luas1 = ContohFungsi(panjang)).

Pada prosedur, nama prosedur tidak bisa digunakan langsung tidak seperti pada Nama Fungsi, yang bisa langsung digunakan dari sebuah prosedur adalah parameternya yang mengandung nilai balik.

Pada dasarnya tidak ada perbedaan yang signifikan antara fungsi dan prosedur pada Bahasa C / C++, hanya dibedakan dari return value. Jika suatu sub program tersebut memiliki nilai balik (return value) maka sub program tersebut disebut Fungsi (Function) namun jika tidak mempunyai nilai balik maka sub program tersebut adalah prosedur (Procedure).

III. Guided

A. Array 1 Dimensi

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

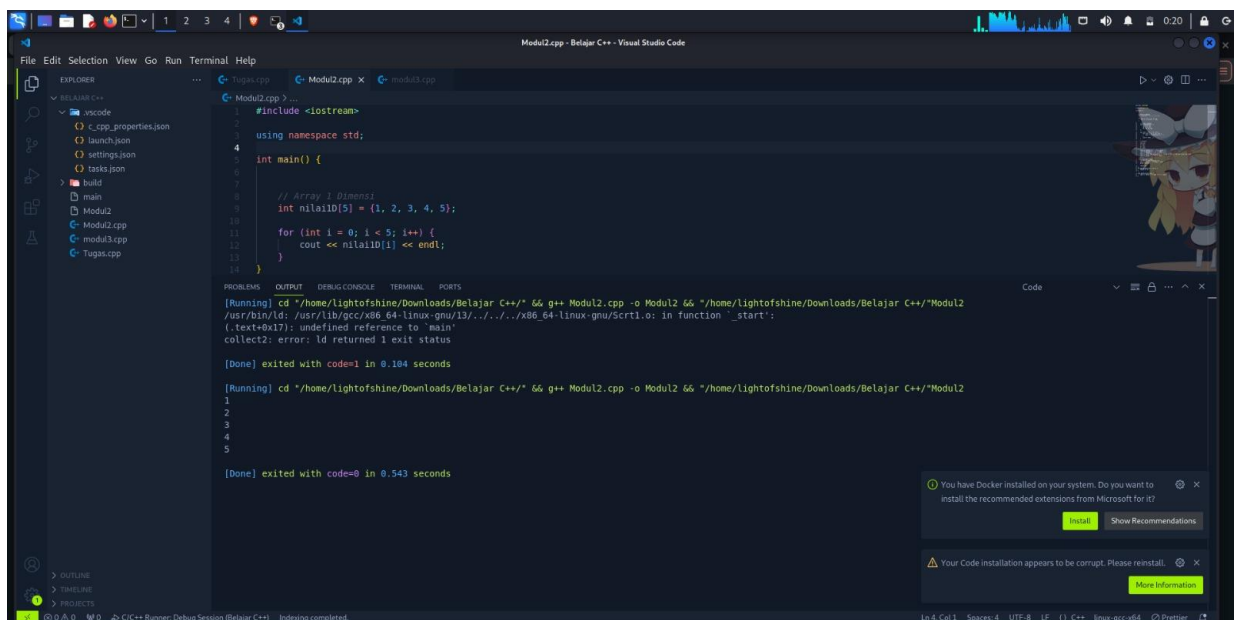
tipe_data nama_var[ukuran]

Keterangan:

Tipe_data : menyatakan jenis elemen array (int, char, float, dll).

Ukuran : menyatakan jumlah maksimum array.

Contoh :



```
Modul2.cpp - Belajar C++ - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  BELAJAR C++
    .vscode
      c_cpp_properties.json
      launch.json
      settings.json
      tasks.json
    build
      main
      Modul2
      Modul2.cpp
      Modul3.cpp
      Tugas.cpp
  Explorer
  Search
  Run and Debug
  Output
  Debug Console
  Terminal
  PROBLEMS
  OUTPUT
  DEBUG CONSOLE
  TERMINAL
  PORTS
  Code

// Array 1 Dimensi
int nilaiD[5] = {1, 2, 3, 4, 5};

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

[Running] cd "/home/lightofshine/Downloads/Belajar C++/" && g++ Modul2.cpp -o Modul2 && "/home/lightofshine/Downloads/Belajar C++/Modul2"
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/13/../../../../x86_64-linux-gnu/libcrti.o: in function '_start':
(.text+0x17): undefined reference to 'main'
collect2: error: ld returned 1 exit status

[Done] exited with code=1 in 0.104 seconds

[Running] cd "/home/lightofshine/Downloads/Belajar C++/" && g++ Modul2.cpp -o Modul2 && "/home/lightofshine/Downloads/Belajar C++/Modul2"
1
2
3
4
5

[Done] exited with code=0 in 0.543 seconds

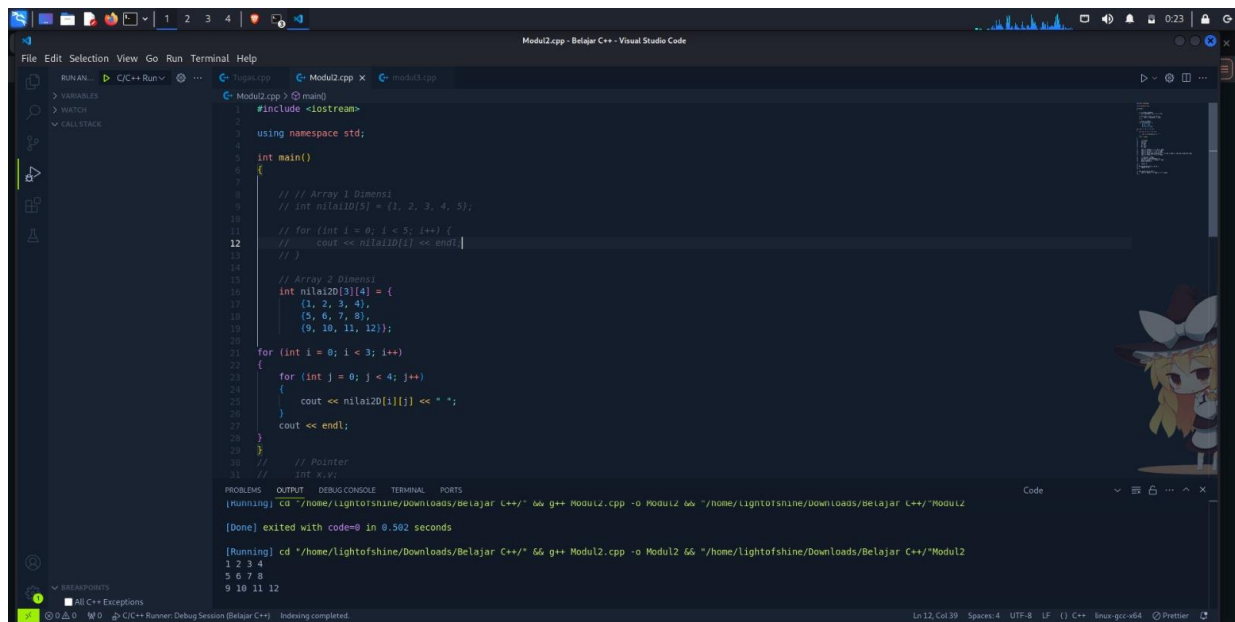
You have Docker installed on your system. Do you want to install the recommended extensions from Microsoft for it?
[Install] Show Recommendations

Your Code installation appears to be corrupt. Please reinstall.
[More Information]
```

B. 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:



```
#include <iostream>
using namespace std;

int main()
{
    // Array 1 Dimensi
    // int nilai1D[5] = {1, 2, 3, 4, 5};

    // for (int i = 0; i < 5; i++) {
    //     cout << nilai1D[i] << endl;
    // }

    // Array 2 Dimensi
    int nilai2D[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 << nilai2D[i][j] << " ";
        }
        cout << endl;
    }

    // Pointer
    // int *p;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

[Running] cd "/home/Lightofshine/Downloads/Belajar C++/" && g++ Modul2.cpp -o Modul2 && "/home/Lightofshine/Downloads/Belajar C++/Modul2

[Done] exited with code=0 in 0.502 seconds

[Running] cd "/home/Lightofshine/Downloads/Belajar C++/" && g++ Modul2.cpp -o Modul2 && "/home/Lightofshine/Downloads/Belajar C++/Modul2

1 2 3 4
5 6 7 8
9 10 11 12

C. 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];

D. 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.

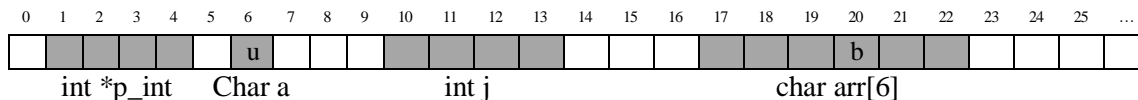
```
p_int = &j;
```

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 = &j; cout<<j<<endl; cout<< &j<< endl; cout<< p_int << endl; cout<< &p_int << endl; cout<< *p_int<<endl; k = *p_int; cout << k << endl;</pre>	<pre>10 X6 X6 X1 10 10</pre>	<p>Nilai variabel j Alamat variabel j Nilai variabel p_int Alamat variabel p_int Nilai variabel yang ditunjuk p_int</p> <p>Nilai variabel k</p>

Contoh :

The screenshot shows a Visual Studio Code editor with a C++ file named 'Modul2.cpp'. The code defines a main function that declares variables x, y, px, and py. It then prints the address of px, the value of x, the value of px (dereferenced), and the value of y. The output window shows the results of these operations.

```
Modul2.cpp - Belajar C++ - Visual Studio Code
File Edit Selection View Go Run Terminal Help
RUN: C/C++ Run...
Modul2.cpp
// for (int i = 0; i < 3; i++)
// cout << nilai2D[i][j] << " ";
// cout << endl;
// }
// }
// Pointer
#include <iostream>
using namespace std;
int main()
{
    int x,y;
    int *px;
    x = 87;
    px = &x;
    y = *px;
    cout << "Alamat x = " << &x << endl;
    cout << "isi px = " << *px << endl;
    cout << "Nilai x = " << x << endl;
    cout << "Nilai yang ditunjuk px = " << *px << endl; // Use *px to get the value
    cout << "Nilai y = " << y << endl;
}
// Function / Fungsi
// int hasil = penjumlahan(5, 3);
// cout << "hasil = " << hasil << endl;
// greet("daniel");
[Running] cd "/home/lightofshine/Downloads/Belajar C++/" && g++ Modul2.cpp -o Modul2 && "/home/lightofshine/Downloads/Belajar C++/Modul2
Alamat x = 0x7ffffac8364d0
isi px = 0x7ffffac8364d0
Nilai x = 87
Nilai yang ditunjuk px = 87
Nilai y = 87
[Done] exited with code=0 in 0.522 seconds
```

E. 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.

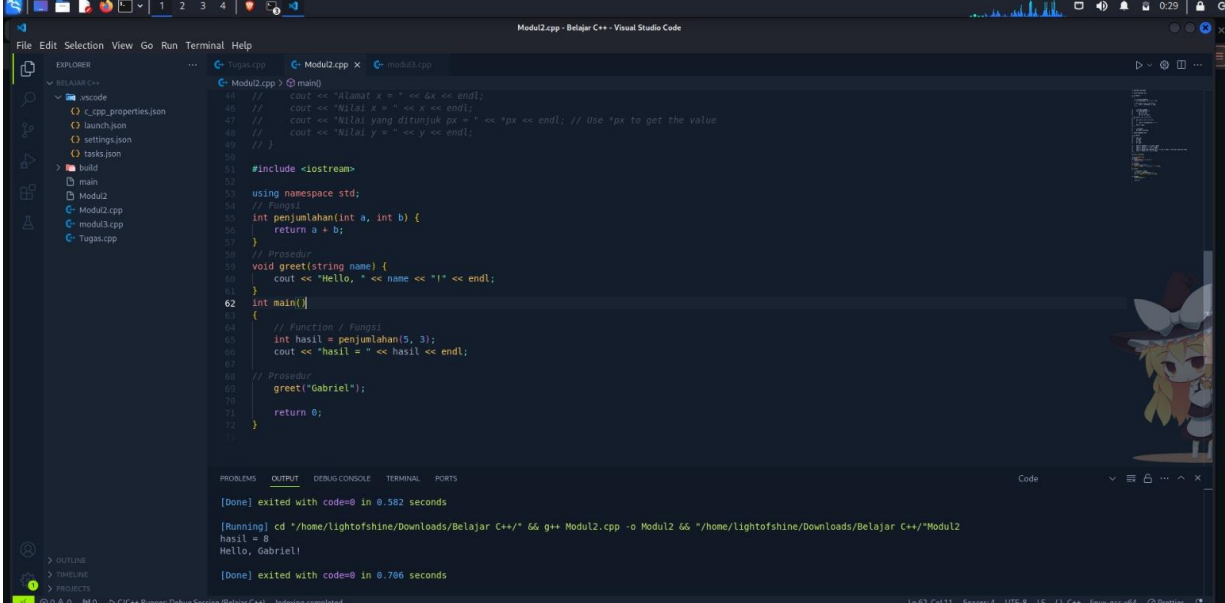
Pada umumnya fungsi memerlukan masukan yang dinamakan sebagai parameter. Masukan ini selanjutnya diolah oleh fungsi. Hasil akhir fungsi berupa sebuah nilai (nilai balik fungsi).

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.

Contoh :



The screenshot shows the Visual Studio Code editor with a C++ file named `Modul2.cpp`. The code defines a function `penjumlahan` and a procedure `greet`. The `main` function calls `penjumlahan` and `greet`. The terminal output shows the program running successfully, displaying the result of the addition and the greeting message.

```
44 // cout << "Alamat x = " << &x << endl;
45 // cout << "Nilai x = " << x << endl;
46 // cout << "Nilai yang ditunjuk pa = " << *pa << endl; // Use *pa to get the value
47 // cout << "Nilai y = " << y << endl;
48 //
49 //
50 //
51
52 #include <iostream>
53
54 using namespace std;
55
56 // Fungsi
57 int penjumlahan(int a, int b) {
58     return a + b;
59 }
60
61 // Prosedur
62 void greet(string name) {
63     cout << "Hello, " << name << "!" << endl;
64 }
65
66 int main()
67 {
68     // Function / Fungsi
69     int hasil = penjumlahan(5, 3);
70     cout << "hasil = " << hasil << endl;
71
72     // Prosedur
73     greet("Gabriel");
74
75     return 0;
76 }
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

[Done] exited with code=0 in 0.582 seconds

[Running] cd "/home/Lightofshine/Downloads/Belajar C++/" && g++ Modul2.cpp -o Modul2 && "/home/Lightofshine/Downloads/Belajar C++/Modul2

hasil = 8
Hello, Gabriel!

[Done] exited with code=0 in 0.786 seconds

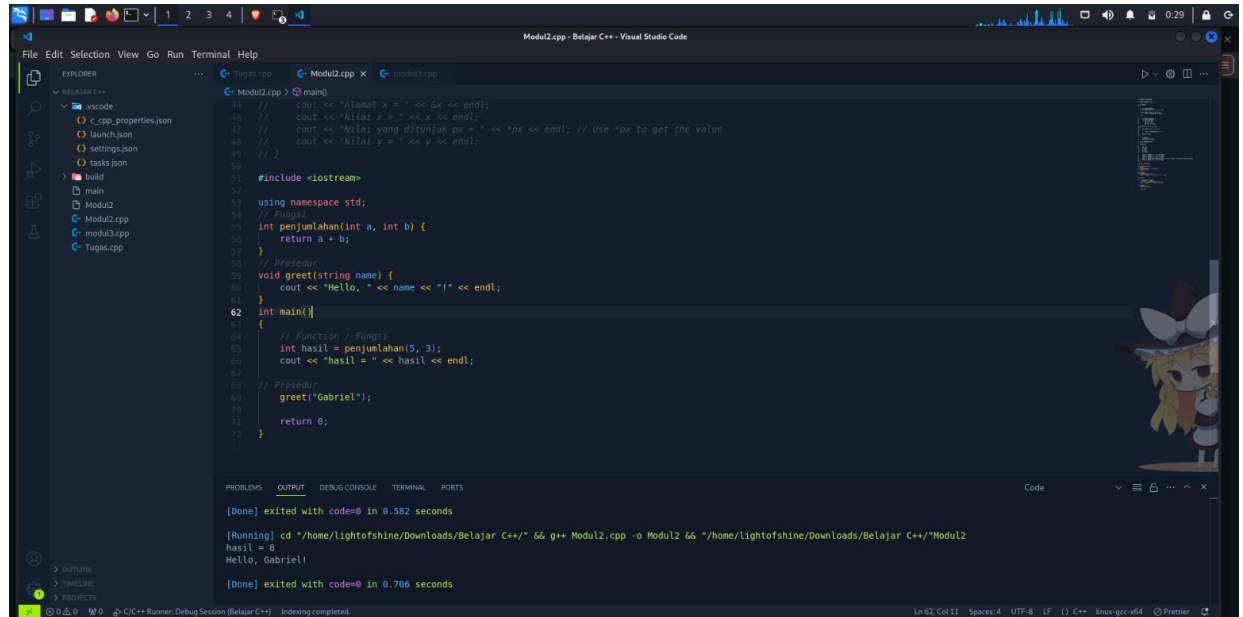
F. Prosedure

Dalam C++ sebenarnya tidak ada prosedur, semua berupa fungsi, termasuk `main()` pun adalah sebuah fungsi. Jadi prosedur dalam C++ merupakan fungsi yang tidak mengembalikan nilai, biasa diawali dengan kata kunci `void` di depan nama prosedur.

Bentuk umum sebuah prosedur :

```
void nama_prosedur (daftar_parameter) {  
    blok pernyataan prosedur ;  
}
```


Contoh :



```
Modu2.cpp - Belajar C++ - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
  BELAJAR C++
    .vscode
    launch.json
    settings.json
    tasks.json
  build
  main
  Modu2
  Modu2.cpp
  Modu3.cpp
  Tugas.cpp

Modu2.cpp
1 //
2 // Alamat x = " << << endl;
3 // cout << "Nilai x = " << x << endl;
4 // cout << "Nilai yang ditunjuk px = " << *px << endl; // Use *px to get the value
5 // cout << "Nilai y = " << y << endl;
6 // }
7
8 #include <iostream>
9
10 using namespace std;
11
12 // Fungsi
13 int penjumlahan(int a, int b) {
14     return a + b;
15 }
16
17 // Prosedur
18 void greet(string name) {
19     cout << "Hello, " << name << "!" << endl;
20 }
21
22 int main()
23 {
24     // Function / Fungsi
25     int hasil = penjumlahan(5, 3);
26     cout << "hasil = " << hasil << endl;
27
28     // Prosedur
29     greet("Gabriel");
30
31     return 0;
32 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[Done] exited with code=0 in 0.582 seconds

[Running] cd "/home/lightofshine/Downloads/Belajar C++/" && g++ Modu2.cpp -o Modu2 && "/home/lightofshine/Downloads/Belajar C++/Modu2
hasil = 8
Hello, Gabriel!

[Done] exited with code=0 in 0.706 seconds

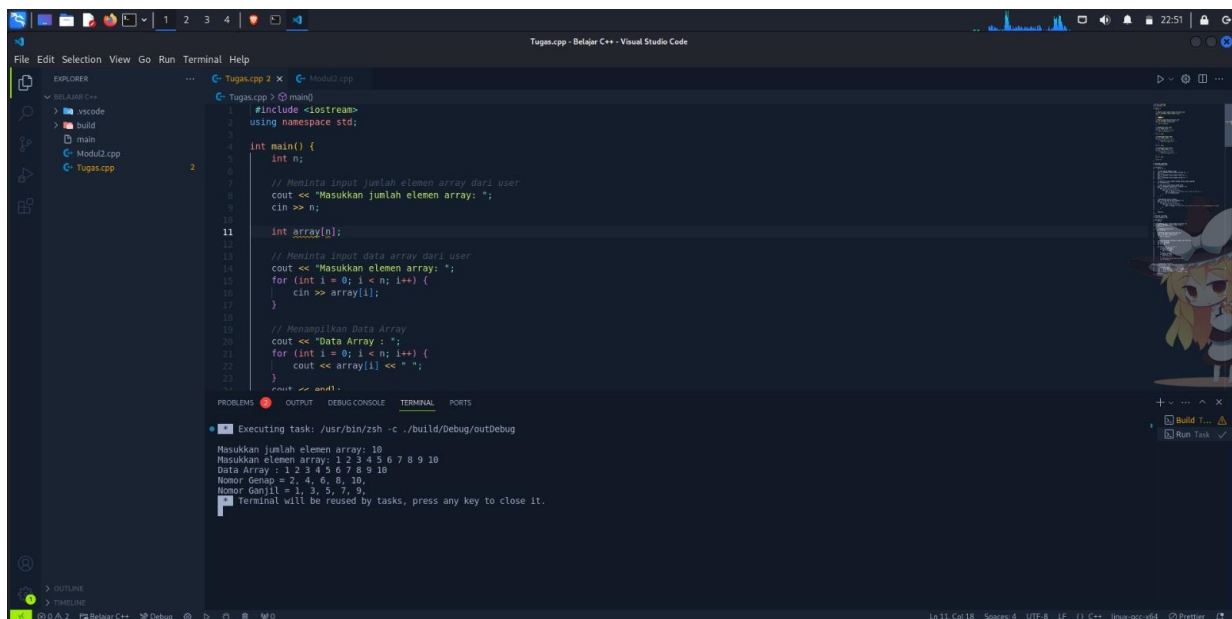
Ln 62, Col 11 Spaces: 4 UTF-8 LF C++ linux-gcc-x64
```

IV. 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,
```

Jawab :



```
Tugas.cpp - Belajar C++ - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
  BELAJAR C++
    .vscode
    build
    main
    Modu2.cpp
    Tugas.cpp

Tugas.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6
7     // Meminta input jumlah elemen array dari user
8     cout << "Masukkan jumlah elemen array: ";
9     cin >> n;
10
11     int array[n];
12
13     // Meminta input data array dari user
14     cout << "Masukkan elemen array: ";
15     for (int i = 0; i < n; i++) {
16         cin >> array[i];
17     }
18
19     // Menampilkan Data Array
20     cout << "Data Array : ";
21     for (int i = 0; i < n; i++) {
22         cout << array[i] << " ";
23     }
24     cout << endl;
25 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Executing task: /usr/bin/zsh -c './build/Debug/outDebug

Masukkan jumlah elemen array: 10
Masukkan elemen array: 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.

Terminal will be reused by tasks, press any key to close it.

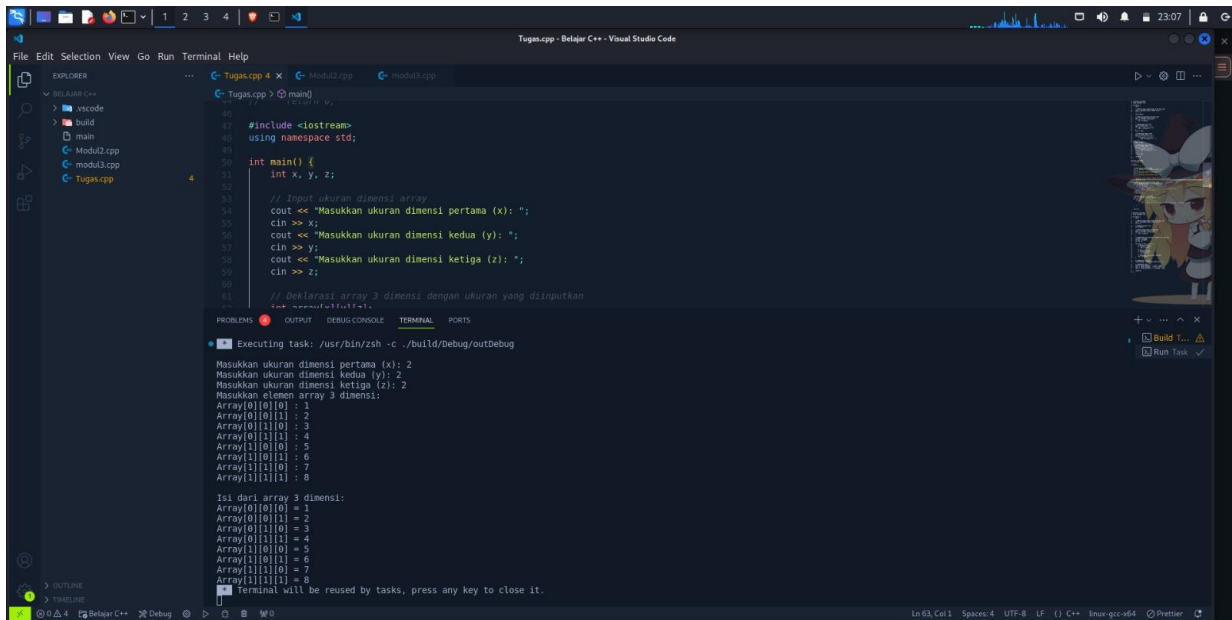
Ln 21, Col 16 Spaces: 4 UTF-8 LF C++ linux-gcc-x64
```



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6
7      // Meminta input jumlah elemen array dari user
8      cout << "Masukkan jumlah elemen array: ";
9      cin >> n;
10
11     int array[n];
12
13     // Meminta input data array dari user
14     cout << "Masukkan elemen array: ";
15     for (int i = 0; i < n; i++) {
16         cin >> array[i];
17     }
18
19     // Menampilkan Data Array
20     cout << "Data Array : ";
21     for (int i = 0; i < n; i++) {
22         cout << array[i] << " ";
23     }
24     cout << endl;
25
26     // Menampilkan Nomor Genap
27     cout << "Nomor Genap = ";
28     for (int i = 0; i < n; i++) {
29         if (array[i] % 2 == 0) {
30             cout << array[i] << ", ";
31         }
32     }
33     cout << endl;
34
35     // Menampilkan Nomor Ganjil
36     cout << "Nomor Ganjil = ";
37     for (int i = 0; i < n; i++) {
38         if (array[i] % 2 != 0) {
39             cout << array[i] << ", ";
40         }
41     }
42     cout << endl;
43
44     return 0;
45 }
```

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

Jawab :

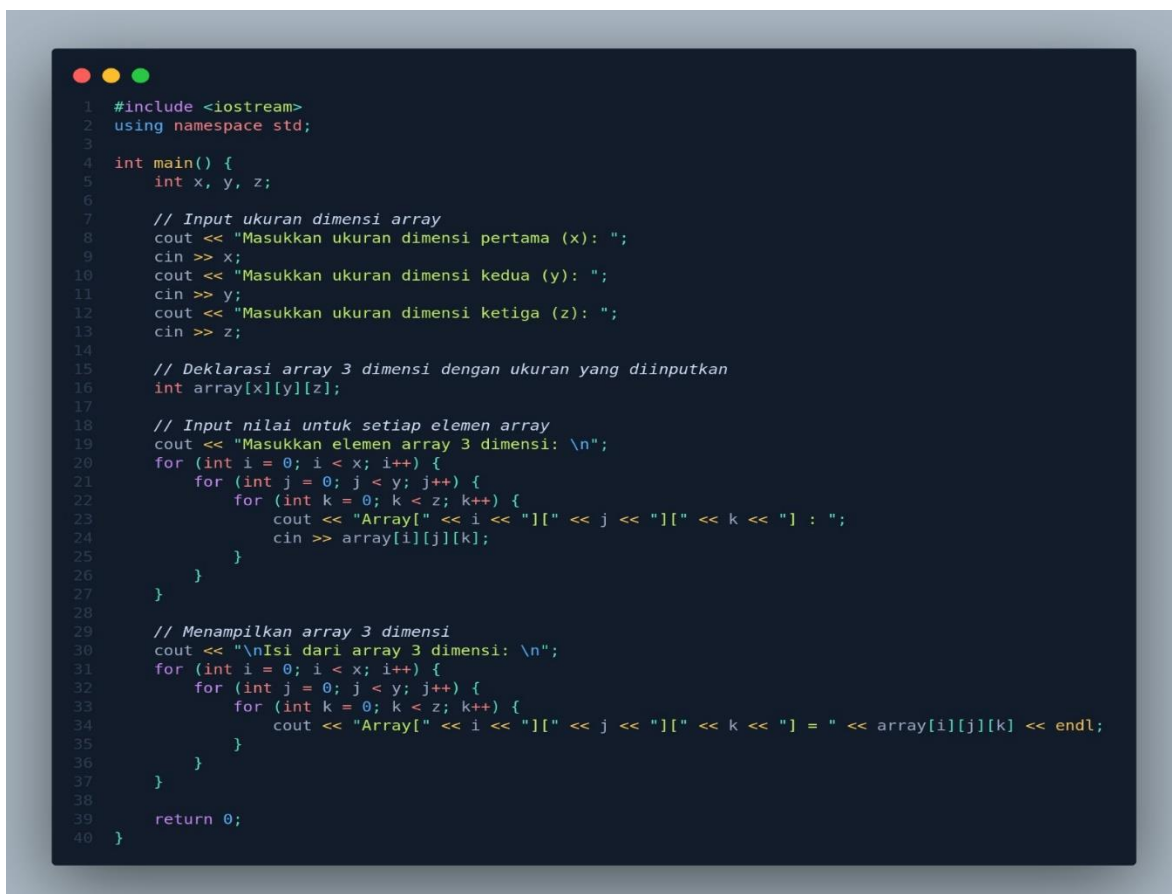


The screenshot shows a Visual Studio Code editor with a C++ file named 'Tugas.cpp'. The code is as follows:

```
1 // Input array 3 dimensi
2 //
3 //
4 #include <iostream>
5 using namespace std;
6
7 int main() {
8     int x, y, z;
9
10    // Input ukuran dimensi array
11    cout << "Masukkan ukuran dimensi pertama (x): ";
12    cin >> x;
13    cout << "Masukkan ukuran dimensi kedua (y): ";
14    cin >> y;
15    cout << "Masukkan ukuran dimensi ketiga (z): ";
16    cin >> z;
17
18    // Deklarasi array 3 dimensi dengan ukuran yang diinputkan
19    int array[x][y][z];
20
21    // Input nilai untuk setiap elemen array
22    for (int i = 0; i < x; i++) {
23        for (int j = 0; j < y; j++) {
24            for (int k = 0; k < z; k++) {
25                cout << "Masukkan elemen array 3 dimensi: ";
26                cin >> array[i][j][k];
27            }
28        }
29    }
30
31    // Menampilkan array 3 dimensi
32    cout << "\nIsi dari array 3 dimensi: \n";
33    for (int i = 0; i < x; i++) {
34        for (int j = 0; j < y; j++) {
35            for (int k = 0; k < z; k++) {
36                cout << "Array[" << i << "][" << j << "][" << k << "] = " << array[i][j][k] << endl;
37            }
38        }
39    }
40    return 0;
41 }
```

The terminal output shows the execution of the program with the following input and output:

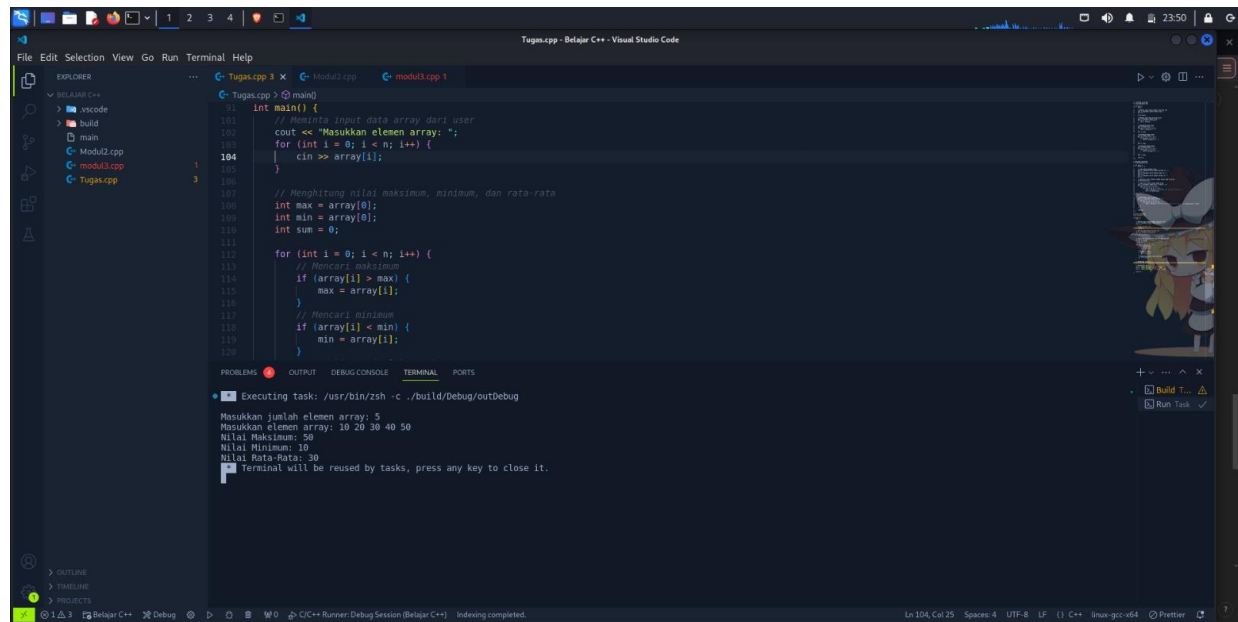
```
Executing Task: /usr/bin/zsh -c ./build/Debug/outDebug
Masukkan ukuran dimensi pertama (x): 2
Masukkan ukuran dimensi kedua (y): 2
Masukkan ukuran dimensi ketiga (z): 2
Masukkan elemen array 3 dimensi:
Array[0][0][0] : 1
Array[0][0][1] : 2
Array[0][1][0] : 3
Array[0][1][1] : 4
Array[1][0][0] : 5
Array[1][0][1] : 6
Array[1][1][0] : 7
Array[1][1][1] : 8
Isi dari array 3 dimensi:
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][1][0] = 7
Array[1][1][1] = 8
Terminal will be reused by tasks, press any key to close it.
```



The screenshot shows a C++ program for inputting and displaying a 3D array. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int x, y, z;
6
7     // Input ukuran dimensi 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    // Deklarasi array 3 dimensi dengan ukuran yang diinputkan
16    int array[x][y][z];
17
18    // Input nilai untuk setiap elemen array
19    cout << "Masukkan elemen array 3 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 << "Array[" << i << "][" << j << "][" << k << "] : ";
24                cin >> array[i][j][k];
25            }
26        }
27    }
28
29    // Menampilkan array 3 dimensi
30    cout << "\nIsi dari array 3 dimensi: \n";
31    for (int i = 0; i < x; i++) {
32        for (int j = 0; j < y; j++) {
33            for (int k = 0; k < z; k++) {
34                cout << "Array[" << i << "][" << j << "][" << k << "] = " << array[i][j][k] << endl;
35            }
36        }
37    }
38
39    return 0;
40 }
```

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



```
1 2 3 4
Tugas.cpp - Belajar C++ - Visual Studio Code
EXPLORER
  RELAY C++
  > vsc
  > build
  > main
  > Modul2.cpp
  > Modul3.cpp
  > Tugas.cpp
1
3
Tugas.cpp
91 int main() {
92     // Meminta input data array dari user
93     cout << "Masukkan elemen array: ";
94     for (int i = 0; i < n; i++) {
95         cin >> array[i];
96     }
97     // Menghitung nilai maksimum, minimum, dan rata-rata
98     int max = array[0];
99     int min = array[0];
100     int sum = 0;
101     for (int i = 0; i < n; i++) {
102         // Mencari maksimum
103         if (array[i] > max) {
104             max = array[i];
105         }
106         // Mencari minimum
107         if (array[i] < min) {
108             min = array[i];
109         }
110         sum += array[i];
111     }
112     // Menghitung rata-rata
113     double avg = sum / n;
114     cout << "Maksimum: " << max << endl;
115     cout << "Minimum: " << min << endl;
116     cout << "Rata-rata: " << avg << endl;
117     return 0;
118 }
```

PROBLEMS OUTPUT DEBUGCONSOLE TERMINAL PORTS

Executing task: /usr/bin/zsh -c ./build/Debug/outDebug

Masukkan jumlah elemen array: 5
Masukkan elemen array: 10 20 30 40 50
Nilai Maksimum: 50
Nilai Minimum: 10
Nilai Rata-Rata: 30
Terminal will be reused by tasks, press any key to close it.

Ln 105, Col 25 Spaces: 4 UTF-8 LF C++ linux-gcc64 Prettier



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6
7      // Meminta input jumlah elemen array dari user
8      cout << "Masukkan jumlah elemen array: ";
9      cin >> n;
10
11     // Mendeklarasikan array
12     int array[n];
13
14     // Meminta input data array dari user
15     cout << "Masukkan elemen array: ";
16     for (int i = 0; i < n; i++) {
17         cin >> array[i];
18     }
19
20     // Menghitung nilai maksimum, minimum, dan rata-rata
21     int max = array[0];
22     int min = array[0];
23     int sum = 0;
24
25     for (int i = 0; i < n; i++) {
26         // Mencari maksimum
27         if (array[i] > max) {
28             max = array[i];
29         }
30         // Mencari minimum
31         if (array[i] < min) {
32             min = array[i];
33         }
34         // Menghitung jumlah untuk rata-rata
35         sum += array[i];
36     }
37
38     // Menghitung rata-rata
39     float average = static_cast<float>(sum) / n;
40
41     // Menampilkan hasil
42     cout << "Nilai Maksimum: " << max << endl;
43     cout << "Nilai Minimum: " << min << endl;
44     cout << "Nilai Rata-Rata: " << average << endl;
45
46     return 0;
47 }
```

V. Kesimpulan

Array, pointer, fungsi, dan prosedur adalah elemen-elemen penting dalam C++ yang memungkinkan kita untuk membangun program yang lebih modular, efisien, dan mudah dipelihara. Array memungkinkan penyimpanan data dalam jumlah besar, sementara pointer memberikan fleksibilitas dalam pengelolaan memori dan manipulasi data. Fungsi dan prosedur membantu kita membagi program menjadi bagian-bagian yang lebih kecil dan logis, meningkatkan pemahaman dan organisasi kode.

Pemahaman yang kuat terhadap konsep-konsep ini sangat penting bagi siapa pun yang Ingin menguasai pemrograman dalam C++. Dengan mengintegrasikan penggunaan array, pointer, fungsi, dan prosedur, kita dapat menulis program yang tidak hanya fungsional tetapi juga efisien dalam hal kinerja dan penggunaan memori.