

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL II
PENGENALAN BAHASA C++**



Disusun Oleh :

NAMA : Besthian Guido Rafael Simbolon

NIM : 103112430258

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

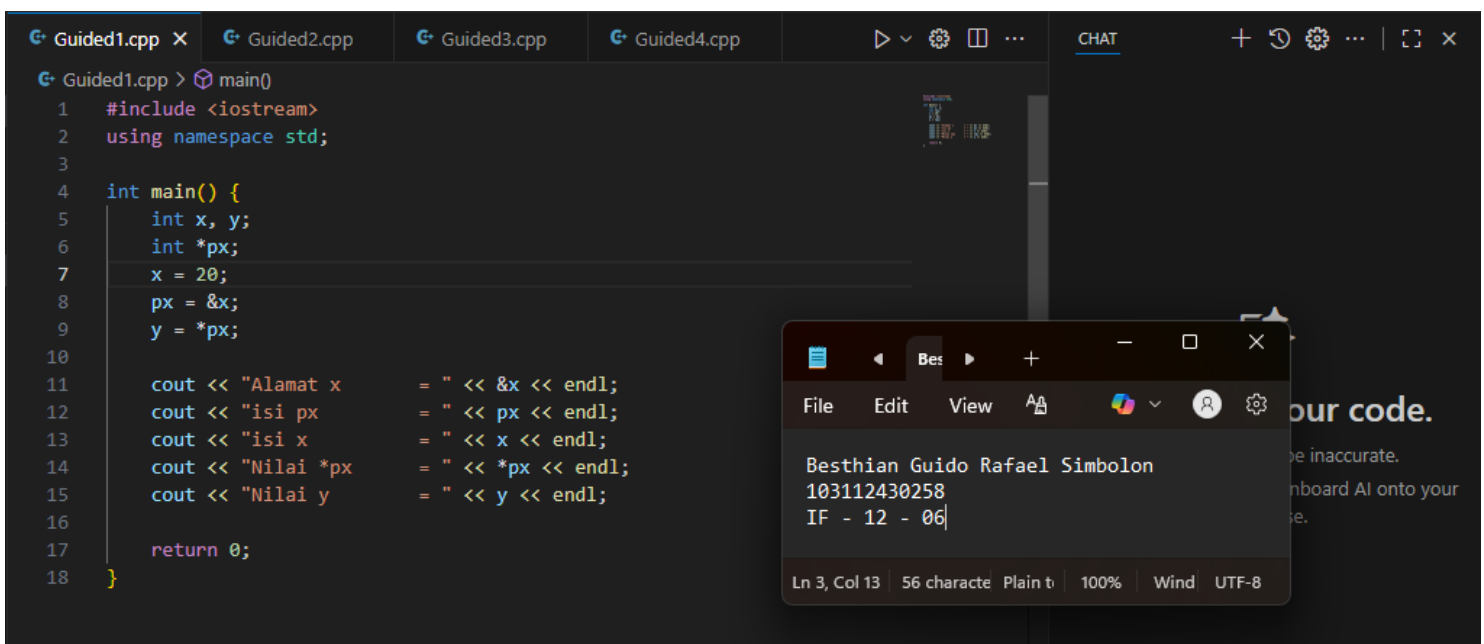
Pada modul ini dibahas empat konsep dasar yang sangat penting dalam pemrograman C++, yaitu array, pointer, fungsi, dan prosedur. Array digunakan untuk menyimpan sekumpulan data dengan tipe yang sama dalam satu variabel, sehingga memudahkan pengelolaan data yang banyak. Setiap elemen array memiliki indeks yang digunakan untuk mengakses datanya. Array dapat berbentuk satu dimensi, dua dimensi, atau bahkan multidimensi, tergantung pada kebutuhan penyimpanan data.

Pointer merupakan variabel yang berfungsi untuk menyimpan alamat memori dari variabel lain. Dengan pointer, program dapat mengakses dan memanipulasi data secara tidak langsung melalui alamat memori, sehingga lebih efisien dalam penggunaan memori dan pengolahan data. Pointer juga sering digunakan bersamaan dengan array karena keduanya saling berhubungan dalam hal akses data di memori.

Fungsi dan prosedur berperan untuk membagi program menjadi bagian-bagian kecil agar lebih terstruktur dan mudah dipahami. Fungsi digunakan untuk melakukan tugas tertentu dan mengembalikan nilai, sedangkan prosedur (fungsi dengan tipe *void*) hanya menjalankan perintah tanpa mengembalikan nilai. Dalam pemanggilan fungsi, dikenal tiga cara pengiriman data, yaitu *call by value*, *call by pointer*, dan *call by reference*. Dengan memahami konsep-konsep ini, seorang programmer dapat menulis program yang lebih efisien, rapi, dan mudah dikembangkan.

B. Guided

Guided 1



```
Guided1.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x, y;
6      int *px;
7      x = 20;
8      px = &x;
9      y = *px;
10
11     cout << "Alamat x      = " << &x << endl;
12     cout << "isi px       = " << px << endl;
13     cout << "isi x        = " << x << endl;
14     cout << "Nilai *px     = " << *px << endl;
15     cout << "Nilai y      = " << y << endl;
16
17     return 0;
18 }
```

Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06

Ln 3, Col 13 | 56 character | Plain text | 100% | Window | UTF-8

```

#include <iostream>
using namespace std;

int main() {
    int x, y;
    int *px;
    x = 20;
    px = &x;
    y = *px;

    cout << "Alamat x      = "
    << &x << endl;
    cout << "isi px      = "
    << px << endl;
    cout << "isi x      = "
    << x << endl;
    cout << "Nilai *px    = "
    << *px << endl;
    cout << "Nilai y      = "
    << y << endl;

    return 0;
}

```

Screenshots Output

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

```

PS C:\StrukDat\Modul 2> cd "c:\StrukDat\Modul 2\" ; if ($?) { g++ Guided1.cpp -o Guided1
Alamat x      = 0xd0429ff800
isi px      = 0xd0429ff800
isi x      = 20
Nilai *px    = 20
Nilai y      = 20
PS C:\StrukDat\Modul 2>

```

Code editor window content:

```

Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06

```

Code editor status bar: Ln 3, Col 13 | 56 character | Plain text | 100% | Window | UTF-8

Deskripsi:

Program di atas merupakan contoh sederhana penggunaan **pointer** dalam bahasa C++. Program ini bertujuan untuk menunjukkan bagaimana sebuah pointer dapat menyimpan alamat dari suatu variabel dan mengakses nilai yang tersimpan di alamat tersebut. Variabel x diberi nilai 20, kemudian alamatnya disimpan di pointer px. Setelah itu, nilai yang ditunjuk oleh pointer (*px) disalin ke variabel y.

Melalui output program, kita dapat melihat bahwa nilai `*px` dan `x` sama, karena `px` menunjuk langsung ke alamat memori tempat `x` disimpan. Program ini membantu memahami konsep dasar pointer, yaitu bagaimana sebuah variabel bisa diakses atau dimanipulasi menggunakan alamat memorinya, bukan hanya melalui namanya secara langsung.

Guided 2

```
Guided2.cpp > main()
1  #include <iostream>
2  #define MAX 5 // Ukuran array maksimal
3
4  using namespace std;
5
6  int main() {
7      int i,j;
8      float nilai[MAX]; // array 1 dimensi
9      static int nilai_tahun[MAX][MAX] = {
10         { 0, 2, 2, 0, 0 },
11         { 0, 1, 1, 1, 0 },
12         { 0, 3, 3, 3, 0 },
13         { 4, 4, 0, 0, 4 },
14         { 5, 0, 0, 0, 5 }
15     };
16
17     // input data array 1 dimensi
18     cout << "=== Input Nilai Siswa ===\n";
19     for (i = 0; i < MAX; i++) {
20         cout << "masukkan nilai ke-" << i + 1
21         << ": ";
22         cin >> nilai[i];
23     }
24
25     // menampilkan isi array 1 dimensi
26     cout << "\n=== Data Nilai Siswa ===\n";
27     for (i = 0; i < MAX; i++) {
28         cout << "Nilai ke-" << i + 1 << " = "
29         << nilai[i]
30         << endl;
31     }
32
33     // menampilkan isi array 2 dimensi
34     cout << "\n=== Nilai Tahunan ===\n";
```

```

#include <iostream>
#define MAX 5 // Ukuran array maksimal

using namespace std;

int main() {
    int i,j;
    float nilai[MAX]; // array 1 dimensi
    static int nilai_tahun[MAX][MAX] = {
        { 0, 2, 2, 0, 0 },
        { 0, 1, 1, 1, 0 },
        { 0, 3, 3, 3, 0 },
        { 4, 4, 0, 0, 4 },
        { 5, 0, 0, 0, 5 }
    };

    // input data array 1 dimensi
    cout << "=== Input Nilai Siswa ===\n";
    for (i = 0; i < MAX; i++) {
        cout << "masukkan nilai ke-" << i + 1
            << ": ";
        cin >> nilai[i];
    }

    // menampilkan isi array 1 dimensi
    cout << "\n=== Data Nilai Siswa ===\n";
    for (i = 0; i < MAX; i++) {
        cout << "Nilai ke-" << i + 1 << " = "
            << nilai[i]
            << endl;
    }

    // menampilkan isi array 2 dimensi
    cout << "\n=== Nilai Tahunan ===\n";
    for (i = 0; i < MAX; i++) {
        for (j = 0; j < MAX; j++) {
            cout << nilai_tahun[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}

```

The image shows a C++ program being executed in a terminal window. The program prompts the user to enter five student scores. The user enters 3, 1, 3, 1, and 8. The program then displays these scores and a 5x5 table of student scores. The table is as follows:

	0	1	2	3	4
0	2	2	0	0	
0	1	1	1	0	
0	3	3	3	0	
4	4	0	0	4	
5	0	0	0	5	

The terminal output is as follows:

```
masukkan nilai ke-3: 3
masukkan nilai ke-4: 1
masukkan nilai ke-5: 8

=== Data Nilai Siswa ===
Nilai ke-1 = 3
Nilai ke-2 = 4
Nilai ke-3 = 3
Nilai ke-4 = 1
Nilai ke-5 = 8

=== Nilai Tahunan ===
0 2 2 0 0
0 1 1 1 0
0 3 3 3 0
4 4 0 0 4
5 0 0 0 5
PS C:\StrukDat\Modul 2>
```

The Notepad++ editor shows the following text:

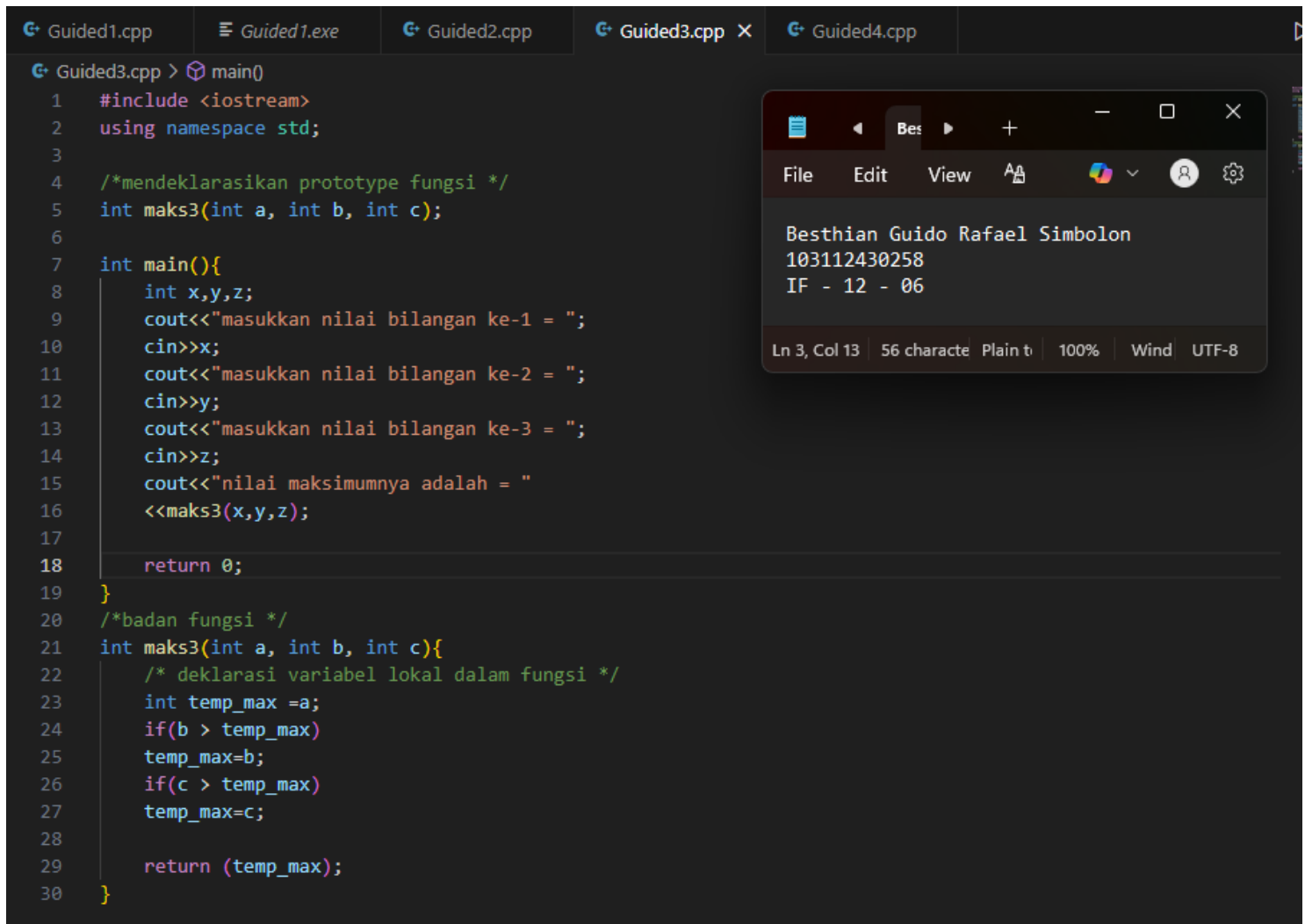
```
Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06
```

Deskripsi

Program di atas bertujuan untuk menunjukkan penggunaan array satu dimensi dan dua dimensi dalam C++. Program ini menggunakan array satu dimensi (nilai) untuk menyimpan lima nilai siswa yang diinput dari pengguna, serta array dua dimensi (nilai_tahun) untuk menyimpan data nilai dalam bentuk tabel berukuran 5x5. Setelah data dimasukkan, program menampilkan kembali nilai yang telah diinput dan juga mencetak isi dari array dua dimensi yang sudah diinisialisasi sebelumnya.

Secara sederhana, program ini membantu memahami cara menyimpan, mengakses, dan menampilkan data dalam array. Array satu dimensi digunakan untuk data sebaris seperti daftar nilai, sedangkan array dua dimensi digunakan untuk data berbentuk tabel, seperti rekap nilai tahunan. Melalui contoh ini, pengguna dapat melihat bagaimana struktur data array mempermudah pengelolaan dan penyajian data yang jumlahnya banyak.

Guided 3



The screenshot shows a C++ IDE with a dark theme. The top bar displays several tabs: Guided1.cpp, Guided1.exe, Guided2.cpp, Guided3.cpp (active), and Guided4.cpp. The main editor window shows the code for Guided3.cpp, with line numbers 1 through 30 on the left. The code defines a function maks3 and a main function that prompts the user for three integers and prints the maximum. An output window on the right shows the program's execution results.

```
1  #include <iostream>
2  using namespace std;
3
4  /*mendeklarasikan prototype fungsi */
5  int maks3(int a, int b, int c);
6
7  int main(){
8      int x,y,z;
9      cout<<"masukkan nilai bilangan ke-1 = ";
10     cin>>x;
11     cout<<"masukkan nilai bilangan ke-2 = ";
12     cin>>y;
13     cout<<"masukkan nilai bilangan ke-3 = ";
14     cin>>z;
15     cout<<"nilai maksimumnya adalah = "
16     <<maks3(x,y,z);
17
18     return 0;
19 }
20 /*badan fungsi */
21 int maks3(int a, int b, int c){
22     /* deklarasi variabel lokal dalam fungsi */
23     int temp_max =a;
24     if(b > temp_max)
25         temp_max=b;
26     if(c > temp_max)
27         temp_max=c;
28
29     return (temp_max);
30 }
```

Output Window:

```
Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06
```

Ln 3, Col 13 | 56 character | Plain text | 100% | Window | UTF-8

```
#include <iostream>

using namespace std;

/*mendeklarasikan prototype fungsi */
int maks3(int a, int b, int c);

int main(){

    int x,y,z;

    cout<<"masukkan nilai bilangan ke-1 = ";

    cin>>x;

    cout<<"masukkan nilai bilangan ke-2 = ";

    cin>>y;

    cout<<"masukkan nilai bilangan ke-3 = ";
```

```

        cin>>z;

        cout<<"nilai maksimumnya adalah = "

        <<maks3(x,y,z);

        return 0;
    }
    /*badan fungsi */
    int maks3(int a, int b, int c){
        /* deklarasi variabel lokal dalam fungsi */
        int temp_max =a;
        if(b > temp_max)
            temp_max=b;
        if(c > temp_max)
            temp_max=c;

        return (temp_max);
    }

```

Screenshot Output

```

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

PS C:\StrukDat\Modul 2> cd "c:\StrukDat\Modul 2\" ; if ($?) { g++ Guided3.cpp -o Guided3 } ; if ($?) { .\Guided3 }
masukkan nilai bilangan ke-1 = 90
masukkan nilai bilangan ke-2 = 39
masukkan nilai bilangan ke-3 = 45
nilai maksimumnya adalah = 90
PS C:\StrukDat\Modul 2> cd "c:\StrukDat\Modul 2\" ; if ($?) { g++ Guided3.cpp -o Guided3 } ; if ($?) { .\Guided3 }
masukkan nilai bilangan ke-1 = 28
masukkan nilai bilangan ke-2 = 40
masukkan nilai bilangan ke-3 = 29
nilai maksimumnya adalah = 40
PS C:\StrukDat\Modul 2>

```

Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06

Ln 3, Col 13 | 56 character | Plain text | 100% | Window | UTF-8

Deskripsi

Program di atas berfungsi untuk menentukan nilai maksimum dari tiga bilangan yang dimasukkan oleh pengguna. Program ini menggunakan konsep fungsi dalam C++, di mana fungsi maks3() dibuat untuk mencari nilai terbesar dari tiga parameter yang diberikan. Di dalam fungsi tersebut, nilai awal temp_max diisi dengan nilai a, lalu dibandingkan dengan b dan c untuk menemukan nilai terbesar di antara ketiganya.

Melalui program ini, kita dapat memahami cara kerja fungsi dalam memecah program menjadi bagian yang lebih kecil dan terstruktur. Fungsi maks3() membantu memisahkan logika perhitungan dari bagian utama program (main()), sehingga kode menjadi lebih rapi, mudah dibaca, dan bisa digunakan kembali jika diperlukan di program lain.

Guided 4



The screenshot shows a C++ IDE with a dark theme. The top bar displays several tabs: Guided1.cpp, Guided1.exe, Guided2.cpp, Guided3.cpp, and Guided4.cpp (which is active). The main editor area shows the code for Guided4.cpp, starting with a main() function. The code includes iostream, uses the std namespace, and defines a tulis() function. The main function prompts the user for the number of lines of text and calls tulis(). The tulis() function is a loop that prints each line number from 1 to x. On the right side, there is a terminal window with a menu bar (File, Edit, View) and a status bar. The terminal displays the name 'Besthian Guido Rafael Simbolon', a student ID '103112430258', and a course 'IF - 12 - 06'. The status bar indicates the cursor is at line 3, column 13, with 56 characters, in plain text, 100% zoom, and UTF-8 encoding.

```
Guided4.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  /*prototype fungsi */
5  void tulis(int x);
6
7  int main(){
8      int jum;
9      cout << " jumlah baris kata=";
10     cin >> jum;
11     tulis(jum);
12     return 0;
13 }
14
15 //badan prosedur
16 void tulis(int x){
17     for (int i = 0; i < x; i++)
18         cout << "baris ke - " << i+1 << endl;
19 }
20
```

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

/*prototype fungsi */
void tulis(int x);

int main(){
    int jum;
```

```

        cout << " jumlah baris
kata=";

        cin >> jum;

        tulis(jum);

        return 0;
}

//badan prosedur
void tulis(int x){
    for (int i = 0; i < x;
i++)
        cout << "baris ke - " <<
i+1 << endl;
}

```

Screensnshot Output

The screenshot shows a terminal window with the following output:

```

PS C:\StrukDat\Modul 2> cd "c:\StrukDat\Modul 2\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?)
{ .\tempCodeRunnerFile }
jumlah baris kata=9
baris ke - 1
baris ke - 2
baris ke - 3
baris ke - 4
baris ke - 5
baris ke - 6
baris ke - 7
baris ke - 8
baris ke - 9
PS C:\StrukDat\Modul 2>

```

An inset window shows a Notepad++ editor with the following text:

```

Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06

```

The status bar at the bottom of the Notepad++ window indicates: Ln 3, Col 13 | 56 character Plain text | 100% | Window UTF-8

Deskripsi

Program di atas merupakan contoh penggunaan prosedur (fungsi void) dalam C++. Program ini meminta pengguna untuk memasukkan jumlah baris yang ingin ditampilkan, lalu mencetak teks “baris ke -” diikuti dengan nomor urutnya sesuai jumlah yang

dimasukkan. Proses pencetakan dilakukan di dalam prosedur tulis(), yang tidak mengembalikan nilai (*void*) tetapi menjalankan perintah tertentu, yaitu menampilkan output ke layar.

Melalui program ini, kita dapat memahami bahwa prosedur digunakan untuk menjalankan tugas tertentu tanpa mengembalikan nilai ke program utama. Dengan memisahkan logika pencetakan ke dalam prosedur tulis(), program menjadi lebih terstruktur, mudah dibaca, dan bisa digunakan kembali jika dibutuhkan di bagian lain dari program.

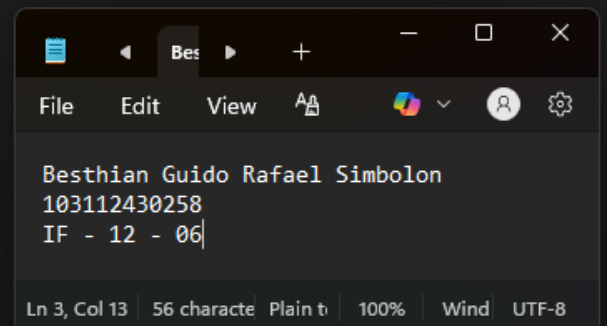
C. Unguided/Tugas

Unguided 1

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string nama[5];
6      float nilai[5][3];
7      float rata[5];
8      int terbaik = 0;
9
10     cout << "=== Input Data Mahasiswa ===" << endl;
11
12     for (int i = 0; i < 5; i++) {
13         cout << "\nNama mahasiswa ke-" << i + 1 << ": ";
14         cin >> nama[i];
15         float total = 0;
16         for (int j = 0; j < 3; j++) {
17             cout << "Nilai mata kuliah ke-" << j + 1 << ": ";
18             cin >> nilai[i][j];
19             total = total + nilai[i][j];
20         }
21         rata[i] = total / 3;
22     }
23
24     for (int i = 1; i < 5; i++) {
25         if (rata[i] > rata[terbaik]) {
26             terbaik = i;
27         }
28     }
29
30     cout << "\n=== Daftar Nilai Mahasiswa ===" << endl;
31     for (int i = 0; i < 5; i++) {
32         cout << "\nNama: " << nama[i] << endl;
33         cout << "Nilai: ";
34         for (int j = 0; j < 3; j++) {

```



```

#include <iostream>
using namespace std;

int main() {
    string nama[5];
    float nilai[5][3];
    float rata[5];
    int terbaik = 0;

    cout << "=== Input
Data Mahasiswa ===" <<
endl;

    for (int i = 0; i <
5; i++) {
        cout << "\nNama
mahasiswa ke-" << i + 1
<< ": ";
        cin >> nama[i];
        float total = 0;
        for (int j = 0; j
< 3; j++) {
            cout <<
"Nilai mata kuliah ke-"
<< j + 1 << ": ";
            cin >>
nilai[i][j];
            total = total
+ nilai[i][j];
        }
        rata[i] = total /
3;
    }

    for (int i = 1; i <
5; i++) {
        if (rata[i] >
rata[terbaik]) {
            terbaik = i;
        }
    }

    cout << "\n=== Daftar
Nilai Mahasiswa ===" <<
endl;
    for (int i = 0; i <
5; i++) {
        cout << "\nNama:
" << nama[i] << endl;
        cout << "Nilai:
";
        for (int j = 0; j
< 3; j++) {
            cout <<
nilai[i][j] << " ";
        }
    }
}

```

```
Unguided1.cpp > main()
1 #include <iostream>

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

Nilai mata kuliah ke-2: 88
Nilai mata kuliah ke-3: 98

Nama mahasiswa ke-4: nadin
Nilai mata kuliah ke-1: 98
Nilai mata kuliah ke-2: 89
Nilai mata kuliah ke-3: 87

Nama mahasiswa ke-5: Simbolon
Nilai mata kuliah ke-1: 100
Nilai mata kuliah ke-2: 100
Nilai mata kuliah ke-3: 100

=== Daftar Nilai Mahasiswa ===

Nama: Sornop
Nilai: 90 87 67
Rata-rata: 81.3333

Nama: Mulyono
Nilai: 69 65 59
Rata-rata: 64.3333

Nama: Baskara
Nilai: 99 88 98
Rata-rata: 95

Nama: nadin
Nilai: 98 89 87
Rata-rata: 91.3333

Nama: Simbolon
Nilai: 100 100 100
Rata-rata: 100 <-- Terbaik
```

Best

File Edit View A A

Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06

Ln 3, Col 13 | 56 character Plain text | 100% | Window UTF-8

Deskripsi:

Program di atas berfungsi untuk menghitung dan menampilkan nilai rata-rata lima mahasiswa serta menentukan siapa yang memiliki rata-rata tertinggi. Program menggunakan array satu dimensi nama untuk menyimpan nama mahasiswa, array dua dimensi nilai untuk menyimpan nilai tiga mata kuliah per mahasiswa, dan array rata untuk menyimpan hasil rata-rata masing-masing mahasiswa. Setelah semua data diinput, program membandingkan setiap nilai rata-rata untuk mencari mahasiswa dengan hasil terbaik.

Secara sederhana, program ini menunjukkan bagaimana array satu dimensi dan dua dimensi dapat digunakan bersama untuk mengelola data yang saling berhubungan. Dengan memanfaatkan perulangan dan struktur data yang teratur, program mampu menampilkan daftar nilai, rata-rata masing-masing mahasiswa, serta menandai mahasiswa dengan rata-rata tertinggi. Program ini menggambarkan penerapan logika dasar pemrograman untuk pengolahan data nilai secara terstruktur dan efisien.

Unguided 2

```
Unguided2.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x;
6      cout << "Masukkan jumlah elemen (N): ";
7      cin >> x;
8
9      int *ptr;
10     ptr = new int[x];
11
12     cout << "\nMasukkan " << x << " bilangan:\n";
13     for (int i = 0; i < x; i++) {
14         cout << "Elemen ke-" << i + 1 << ": ";
15         cin >> *(ptr + i);
16     }
17
18     int total = 0;
19     int maks = *ptr;
20     int min = *ptr;
21
22     for (int i = 0; i < x; i++) {
23         int nilai = *(ptr + i);
24         total = total + nilai;
25
26         if (nilai > maks)
27             maks = nilai;
28         if (nilai < min)
29             min = nilai;
30     }
31
32     cout << "\n=== Hasil Perhitungan ===" << endl;
33     cout << "Jumlah semua bilangan = " << total << endl;
34     cout << "Nilai maksimum = " << maks << endl;
35 }
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int x;
```

```
    cout << "Masukkan jumlah elemen (N): ";
```

```
    cin >> x;
```

```
    int *ptr;
```

```
ptr = new int[x];

cout << "\nMasukkan " << x << " bilangan:\n";
for (int i = 0; i < x; i++) {
    cout << "Elemen ke-" << i + 1 << ": ";
    cin >> *(ptr + i);
}

int total = 0;
int maks = *ptr;
int min = *ptr;

for (int i = 0; i < x; i++) {
    int nilai = *(ptr + i);
    total = total + nilai;

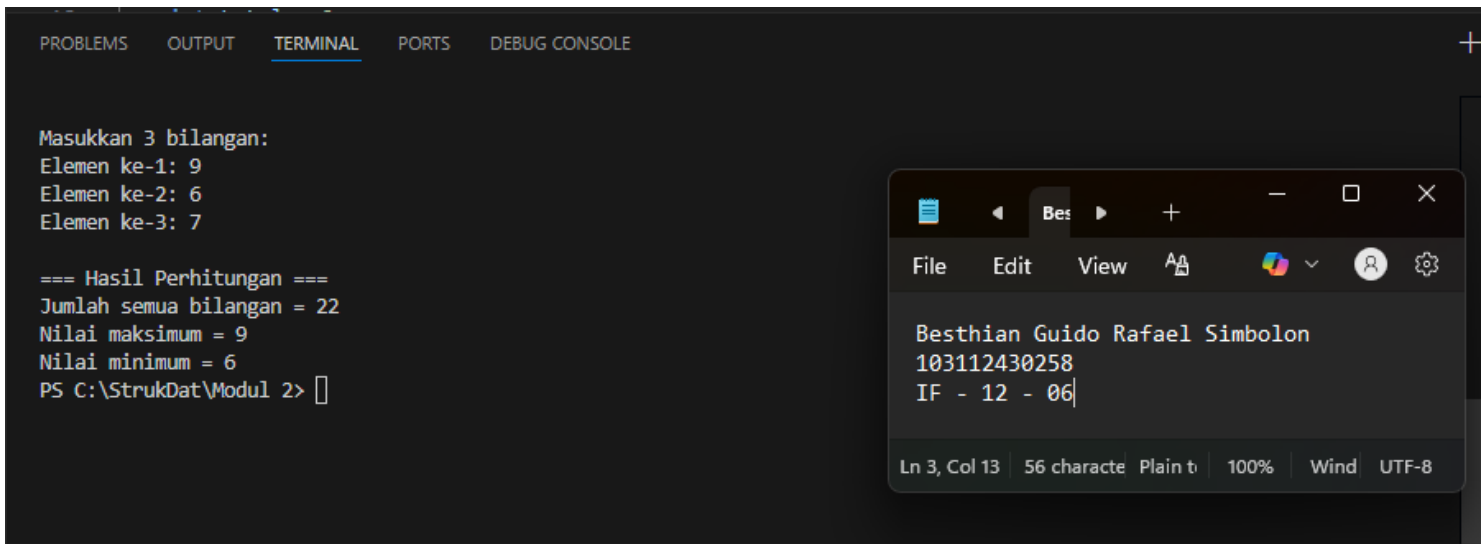
    if (nilai > maks)
        maks = nilai;
    if (nilai < min)
        min = nilai;
}

cout << "\n=== Hasil Perhitungan ===" << endl;
cout << "Jumlah semua bilangan = " << total << endl;
cout << "Nilai maksimum = " << maks << endl;
cout << "Nilai minimum = " << min << endl;

delete[] ptr;

return 0;
}
```

Screenshot Output



The screenshot shows a terminal window with the following output:

```
Masukkan 3 bilangan:  
Elemen ke-1: 9  
Elemen ke-2: 6  
Elemen ke-3: 7  
  
=== Hasil Perhitungan ===  
Jumlah semua bilangan = 22  
Nilai maksimum = 9  
Nilai minimum = 6  
PS C:\StrukDat\Modul 2> []
```

Overlaid on the terminal is a Notepad++ window titled 'Best' containing the following text:

```
Besthian Guido Rafael Simbolon  
103112430258  
IF - 12 - 06
```

The Notepad++ status bar at the bottom indicates: Ln 3, Col 13 | 56 character Plain text | 100% | Window UTF-8.

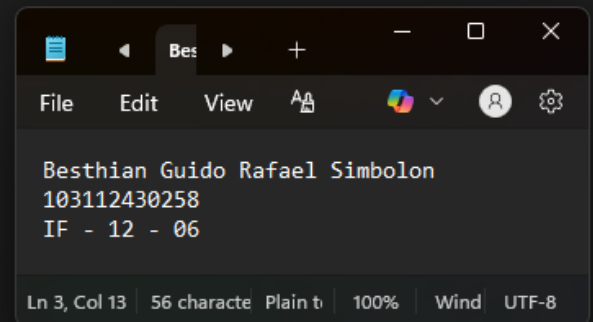
Deksripsi

Program di atas berfungsi untuk mengolah sejumlah bilangan yang dimasukkan pengguna menggunakan konsep pointer dan alokasi memori dinamis. Program meminta pengguna menentukan jumlah elemen yang akan diinput, lalu menggunakan pointer `ptr` untuk membuat array dinamis dengan perintah `new int[x]`. Setiap bilangan disimpan melalui pointer, kemudian dihitung totalnya, serta dicari nilai maksimum dan minimum dari semua bilangan yang dimasukkan.

Melalui program ini, kita dapat memahami cara menggunakan pointer untuk mengakses dan mengelola data di memori secara dinamis. Konsep ini sangat berguna ketika jumlah data belum diketahui di awal program. Setelah proses selesai, memori yang digunakan dilepaskan dengan perintah `delete[] ptr` agar tidak terjadi pemborosan memori (*memory leak*). Program ini menjadi contoh sederhana penerapan pointer dalam manajemen memori dan pengolahan data numerik.

Unguided 3

```
Unguided3.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  float hitungRata(int nilai[], int x) {
5      float total = 0;
6      for (int i = 0; i < x; i++) {
7          total = total + nilai[i];
8      }
9      return total / x;
10 }
11
12 void cariNilai(int nilai[], int x, int &maks, int &min) {
13     maks = nilai[0];
14     min = nilai[0];
15     for (int i = 1; i < x; i++) {
16         if (nilai[i] > maks)
17             maks = nilai[i];
18         if (nilai[i] < min)
19             min = nilai[i];
20     }
21 }
22
23 int main() {
24     int siswa;
25     cout << "Masukkan jumlah siswa: ";
26     cin >> siswa;
27
28     int nilai[siswa];
29     for (int i = 0; i < siswa; i++) {
30         cout << "Nilai siswa ke-" << i + 1 << ": ";
31         cin >> nilai[i];
32     }
33
34     float rata = hitungRata(nilai, siswa);
```



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

float hitungRata(int nilai[], int x) {
    float total = 0;
    for (int i = 0; i < x; i++) {
        total = total + nilai[i];
    }
    return total / x;
}

void cariNilai(int nilai[], int x, int &maks, int &min) {
    maks = nilai[0];
    min = nilai[0];
    for (int i = 1; i < x; i++) {
```

```

        if (nilai[i] > maks)
            maks = nilai[i];
        if (nilai[i] < min)
            min = nilai[i];
    }
}

int main() {
    int siswa;
    cout << "Masukkan jumlah siswa: ";
    cin >> siswa;

    int nilai[siswa];
    for (int i = 0; i < siswa; i++) {
        cout << "Nilai siswa ke-" << i + 1 << ": ";
        cin >> nilai[i];
    }

    float rata = hitungRata(nilai, siswa);

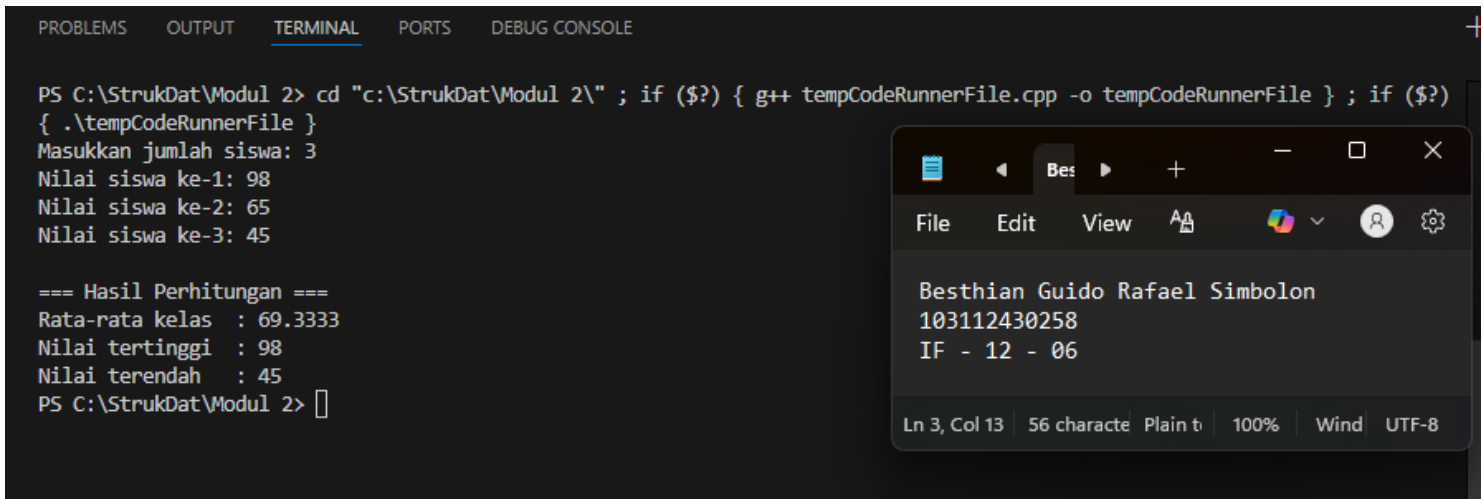
    int tertinggi, terendah;
    cariNilai(nilai, siswa, tertinggi, terendah);

    cout << "\n=== Hasil Perhitungan ===" << endl;
    cout << "Rata-rata kelas   : " << rata << endl;
    cout << "Nilai tertinggi    : " << tertinggi << endl;
    cout << "Nilai terendah     : " << terendah << endl;

    return 0;
}

```

Screenshot Output



The screenshot shows a Windows terminal window with a dark background. The terminal output is as follows:

```
PS C:\StrukDat\Modul 2> cd "c:\StrukDat\Modul 2\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan jumlah siswa: 3
Nilai siswa ke-1: 98
Nilai siswa ke-2: 65
Nilai siswa ke-3: 45

=== Hasil Perhitungan ===
Rata-rata kelas : 69.3333
Nilai tertinggi : 98
Nilai terendah : 45
PS C:\StrukDat\Modul 2> 
```

Overlaid on the right side of the terminal is a small, semi-transparent window titled 'Besthian Guido Rafael Simbolon'. It contains the text '103112430258' and 'IF - 12 - 06'. Below this text, it shows 'Ln 3, Col 13', '56 character', 'Plain text', '100%', 'Wind', and 'UTF-8'.

Dekripsi

Program di atas berfungsi untuk menghitung rata-rata, nilai tertinggi, dan nilai terendah dari sejumlah nilai siswa menggunakan konsep fungsi dan parameter dalam C++. Program menggunakan fungsi `hitungRata()` untuk menghitung nilai rata-rata seluruh siswa dengan menjumlahkan semua elemen array, kemudian membaginya dengan jumlah siswa. Fungsi `cariNilai()` digunakan untuk mencari nilai maksimum dan minimum dengan melewati parameter secara *by reference*, sehingga nilai hasil perhitungan dapat langsung dikembalikan ke variabel di program utama.

Melalui program ini, kita dapat memahami bagaimana fungsi membantu membagi program menjadi bagian yang lebih terstruktur dan efisien. Fungsi `hitungRata()` dan `cariNilai()` membuat kode lebih rapi, mudah dibaca, serta menghindari pengulangan logika yang sama di dalam program. Program ini juga menunjukkan penerapan parameter *by reference* untuk mengubah nilai variabel langsung dari dalam fungsi tanpa perlu mengembalikan nilai secara eksplisit.

Unguided 4

```
Unguided4.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  void tampilSegitiga(int uk) {
5      for (int i = 1; i <= uk; i++) {
6          for (int j = 1; j <= i; j++) {
7              cout << j << " ";
8          }
9          cout << endl;
10     }
11 }
12
13 int main() {
14     int x;
15     cout << "Masukkan nilai x: ";
16     cin >> x;
17
18     cout << "\nPola segitiga angka:\n";
19     tampilSegitiga(x);
20
21     return 0;
22 }
```

File Edit View A A

Best

Besthian Guido Rafael Simbolon
103112430258
IF - 12 - 06Z

Ln 3, Col 14 | 57 character | Plain text | 100% | Window | UTF-8

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

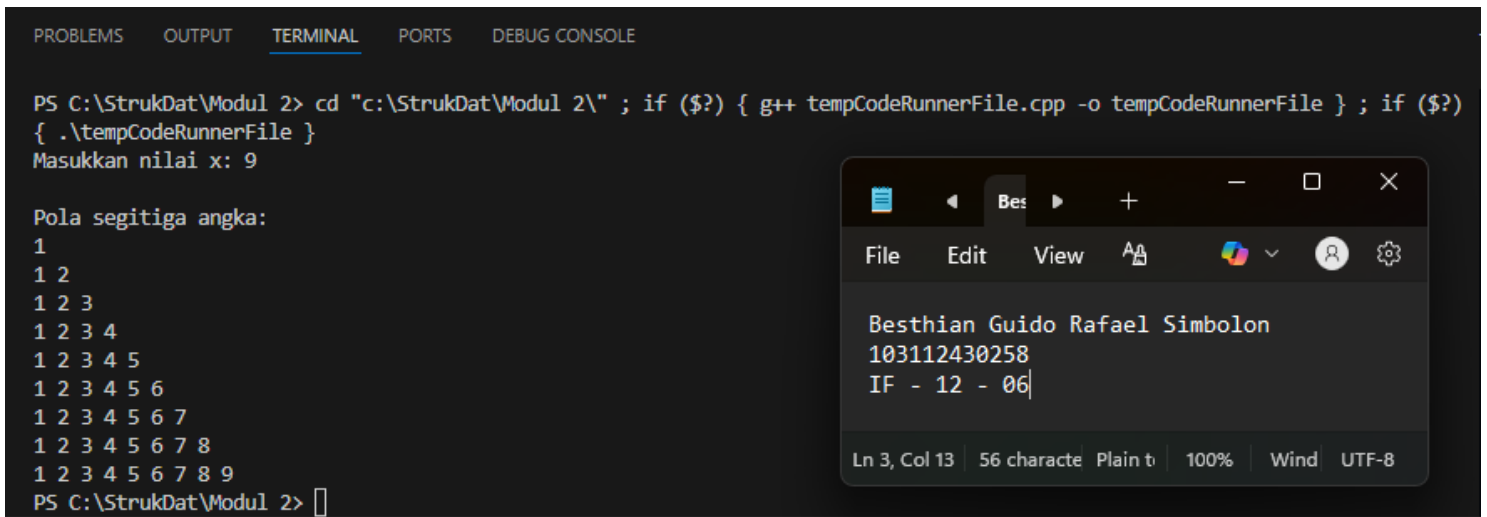
void tampilSegitiga(int uk) {
    for (int i = 1; i <= uk; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
}

int main() {
    int x;
    cout << "Masukkan nilai x: ";
    cin >> x;

    cout << "\nPola segitiga angka:\n";
    tampilSegitiga(x);

    return 0;
}
```

Screenshot Output



```
PS C:\StrukDat\Modul 2> cd "c:\StrukDat\Modul 2\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan nilai x: 9

Pola segitiga angka:
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
PS C:\StrukDat\Modul 2> 
```

Deksripsi

Program di atas berfungsi untuk menampilkan pola segitiga angka berdasarkan input pengguna dengan menggunakan konsep fungsi dan perulangan bersarang (nested loop). Pengguna diminta memasukkan sebuah nilai x yang menentukan tinggi segitiga. Fungsi tampilSegitiga() kemudian mencetak angka secara berurutan mulai dari 1 hingga nilai i pada setiap baris, sehingga membentuk pola segitiga yang bertambah setiap barisnya.

Melalui program ini, kita dapat memahami cara menggunakan fungsi untuk memisahkan logika tampilan dari program utama serta penerapan perulangan bersarang dalam mencetak pola tertentu. Program ini sederhana namun efektif untuk melatih pemahaman tentang struktur kendali for dan penggunaan fungsi dalam membuat program yang lebih terorganisir.

D. Kesimpulan

Dari seluruh program yang telah dibuat, dapat disimpulkan bahwa pemahaman terhadap array, pointer, fungsi, dan prosedur sangat penting dalam pemrograman C++. Array digunakan untuk menyimpan banyak data dengan tipe yang sama agar pengolahan data menjadi lebih mudah dan terstruktur. Pointer berfungsi untuk mengakses dan mengelola data melalui alamat memori, sehingga penggunaan memori menjadi lebih efisien.

Selain itu, penggunaan fungsi dan prosedur membantu memecah program menjadi bagian-bagian kecil yang memiliki tugas tertentu. Hal ini membuat program lebih rapi, mudah dibaca, dan lebih mudah diperbaiki ketika terjadi kesalahan. Melalui berbagai

contoh program seperti perhitungan nilai, pencarian data, dan pembuatan pola, dapat dipahami bahwa kombinasi antara array, pointer, serta fungsi dan prosedur sangat berguna untuk membuat program yang terstruktur, efisien, dan mudah dikembangkan.

.

Referensi

Juhana, A. (n.d.). *Algoritma dan Pemrograman Dasar Menggunakan Bahasa C++*. Rumah Publikasi Indonesia.

Supardi. (2015). *Belajar Pemrograman C++ untuk Pemula*. Jakarta: Elex Media Komputindo.

Wahyudi, B. (2018). *Struktur Data dan Implementasinya dalam Bahasa C++*. Bandung: Informatika.

Budiman, A. (2021). *Logika dan Algoritma Pemrograman Menggunakan C++*. Yogyakarta: Deepublish.

Kadir, A. (2019). *Dasar Pemrograman Komputer Menggunakan C/C++*. Yogyakarta: Andi Publisher.

Pratama, R. (2020). *Pemrograman Terstruktur dengan Bahasa C++*. Surabaya: Scopindo Media Pustaka.