

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL II
PENGENALAN BAHASA C++
(BAGIAN KEDUA)**



Disusun Oleh :

Nama : Izzah Minkhotun Fannisa

NIM : 103112400198

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

C++ merupakan bahasa pemrograman yang pertama kali dikembangkan oleh Bjarne Stroustrup, seorang ilmuwan komputer asal Denmark. Bahasa pemrograman C++ disempurnakan dari bahasa C yang di temukan oleh Dennis Ritchie pada awal 1970-an. Lalu resmi dirilis pada tahun 1985. C++ cukup banyak digunakan karena kemampuannya dalam memberikan kecepatan eksekusi tinggi serta dukungan terhadap manajemen memori tingkat rendah.

B. Guide

1. Guide 1

a. Source Code

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

int main() {
    int x, y;
    int *px;
    x = 87;
    px = &x;
    y = *px;

    // Menampilkan informasi
    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;
}
```

b. Screenshot Output

```
Alamat x =0x5ffee4
Isi px =0x5ffee4
Isi x =87
Nilai *px =87
Nilai y =87
PS C:\Users\ACER> □
```

c. Deskripsi Program

Program tersebut digunakan untuk memperlihatkan cara kerja pointer dalam bahasa C++. Pada program ini, sebuah variabel x diinisialisasi dengan nilai 87, kemudian sebuah pointer px digunakan untuk menyimpan alamat memori dari variabel x melalui perintah `px = &x`. Selanjutnya, nilai yang disimpan di alamat tersebut diambil menggunakan operator dereferensi `*px` dan disalin ke variabel y. Program menampilkan lima informasi utama, yaitu alamat

variabel x, isi pointer px (yang sama dengan alamat x), isi variabel x, nilai dari *px, serta nilai y. Melalui tampilan tersebut, dapat disimpulkan bahwa pointer menyimpan alamat memori suatu variabel, dan dengan operator (*) kita dapat mengakses nilai yang ada di alamat tersebut.

2. Guide 2

a. Source Code

```
#include <iostream> //array
#define MAX 5 // UKURAN ARRAY DITETAPKAN SEBAGAI

using namespace std;

int main() {
    int i, j;
    float nilai[MAX]; // Array 1 dimensi
    static int nilai_tahun[MAX][MAX] = { // nilai tetap
        {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 (perulangan for)
    cout << "\n=== Data 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;
}
}

```

b. Screenshot Output

```

=== Data Nilai Siswa ===
Masukkan Nilai ke-1: 100
Masukkan Nilai ke-2: 90
Masukkan Nilai ke-3: 70
Masukkan Nilai ke-4: 75
Masukkan Nilai ke-5: 86

=== Data Nilai Siswa ===
Nilai ke-1 = 100
n=== 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

```

c. Deskripsi Program

Program ini digunakan untuk memperlihatkan penggunaan array satu dimensi dan array dua dimensi dalam bahasa C++. Program mendeklarasikan konstanta MAX bernilai 5 sebagai ukuran tetap array. Array satu dimensi nilai[MAX] digunakan untuk menyimpan input nilai dari pengguna, sedangkan array dua dimensi nilai_tahun[MAX][MAX] telah diisi secara statis dengan nilai-nilai tetap yang merepresentasikan data nilai tahunan. Program pertama-tama meminta pengguna untuk memasukkan lima nilai ke dalam array satu dimensi menggunakan perulangan for, kemudian menampilkan kembali data tersebut ke layar. Setelah itu, program menampilkan isi array dua dimensi nilai_tahun dalam bentuk tabel dengan bantuan dua perulangan bersarang. Hasil akhir program menunjukkan bagaimana data dapat disimpan dan ditampilkan dalam bentuk array 1 dimensi (sebaris) dan array 2 dimensi (berbentuk matriks).

3. Guide 3

a. Source Code

```

#include <iostream> // input output program
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;
}

```

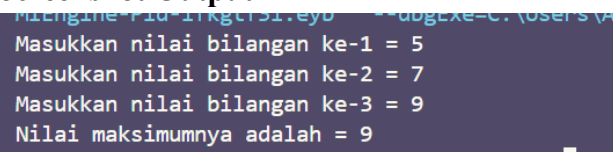
```

        cout << "Nilai maksimumnya adalah = "
        << maks3 (x, y, z);
        return 0;
    }

    int maks3(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;
    }
}

```

b. Screenshot Output



```

Microsoft Windows [C:\Users\A...
Masukkan nilai bilangan ke-1 = 5
Masukkan nilai bilangan ke-2 = 7
Masukkan nilai bilangan ke-3 = 9
Nilai maksimumnya adalah = 9

```

c. Deskripsi Program

Program tersebut berfungsi untuk mencari nilai terbesar dari tiga bilangan bulat yang dimasukkan oleh pengguna. Program menerapkan konsep fungsi dengan parameter formal untuk membandingkan ketiga nilai tersebut dan mengembalikan nilai tertinggi. Pada bagian awal, fungsi maks3() dideklarasikan sebagai prototype dan kemudian dipanggil di dalam fungsi main(). Fungsi maks3() menerima tiga parameter (a, b, dan c), lalu menggunakan percabangan if untuk menentukan bilangan yang memiliki nilai paling besar. Nilai sementara disimpan dalam variabel temp_max sebelum hasil akhirnya dikembalikan ke fungsi main() dan ditampilkan di layar.

4. Guide 4

a. Source Code

```

#include <iostream>
using namespace std;

// Prototype Prosedur
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;
    }
}
```

b. Screenshot Output

```
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
```

c. Deskripsi Program

Program ini merupakan contoh sederhana penggunaan prosedur dalam bahasa C++. Program meminta pengguna memasukkan jumlah baris yang ingin dicetak, kemudian memanggil prosedur tulis() yang menerima jumlah tersebut sebagai parameter. Di dalam prosedur, digunakan perulangan for untuk menampilkan setiap baris secara berurutan dengan format "Baris ke-1", "Baris ke-2", dan seterusnya hingga jumlah yang diminta. Konsep yang diperlihatkan mencakup penerimaan input dari pengguna, pemanggilan prosedur dengan parameter, dan penggunaan looping untuk mengeksekusi perintah berulang. Program ini bermanfaat untuk memahami bagaimana fungsi/prosedur dapat memisahkan logika tertentu dari fungsi utama (main) sehingga kode lebih terstruktur dan mudah dibaca.

C. Unguide

1. Unguide 1

a. Source Code

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

int main() {
    const int MAHASISWA = 5;
    const int MATAKULIAH = 3;

    string nama[MAHASISWA];
    float nilai[MAHASISWA][MATAKULIAH];
    float rata[MAHASISWA];
    int terbaik = 0;
```

```

cout << "Nilai Mahasiswa \n";
cout << "===== \n";

for (int i = 0; i < MAHASISWA; i++) {
    cout << "\nNama Mahasiswa ke-" << i + 1 << " : ";
    cin >> nama[i];

    float total = 0;
    for (int j = 0; j < MATAKULIAH; j++) {
        cout << " Nilai Mata kuliah ke-" << j + 1 << " : ";
        cin >> nilai[i][j];
        total += nilai[i][j];
    }
    rata[i] = total / MATAKULIAH;
}

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

cout << "\n\nDaftar Nilai Mahasiswa\n";
cout <<
"===== \n";
cout << left << setw(12) << "Nama"
<< setw(10) << "Mata Kuliah 1"
<< setw(10) << "Mata Kuliah 2"
<< setw(10) << "Mata Kuliah 3"
<< setw(10) << "Rata-Rata"
<< "Keterangan \n";
cout << "-----
--\n";

for (int i = 0; i < MAHASISWA; i++) {
    cout << left << setw(12) << nama[i];
    for (int j = 0; j < MATAKULIAH; j++) {
        cout << setw(10) << nilai[i][j];
    }
    cout << setw(10) << fixed << setprecision(2) << rata[i];
    if (i == terbaik)
        cout << "Terbaik";
    cout << endl;
}
cout <<
"===== \n";

```

```

        cout << "Mahasiswa dengan rata-rata tertinggi: " <<
nama[terbaik]
        << " (" << rata[terbaik] << ")\n";

        return 0;
}

```

b. Screenshot Output

The screenshot shows the output of a C++ program. It first displays the names and scores of five students across three subjects. Then, it prints a table summarizing the data, including the highest average score and the student with the highest average score.

```

G: arraycpp > ...
1 #include <iostream>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Nilai Mahasiswa
=====
Nama Mahasiswa ke-1 : Kahitna
Nilai Mata kuliah ke-1 : 75.5
Nilai Mata kuliah ke-2 : 85
Nilai Mata kuliah ke-3 : 90

Nama Mahasiswa ke-2 : Lanara
Nilai Mata kuliah ke-1 : 89
Nilai Mata kuliah ke-2 : 95.1
Nilai Mata kuliah ke-3 : 92.3

Nama Mahasiswa ke-3 : Candelaria
Nilai Mata kuliah ke-1 : 98
Nilai Mata kuliah ke-2 : 87
Nilai Mata kuliah ke-3 : 90

Nama Mahasiswa ke-4 : Lamie
Nilai Mata kuliah ke-1 : 78
Nilai Mata kuliah ke-2 : 90
Nilai Mata kuliah ke-3 : 97

Nama Mahasiswa ke-5 : Reverie
Nilai Mata kuliah ke-1 : 91
Nilai Mata kuliah ke-2 : 97
Nilai Mata kuliah ke-3 : 100

Daftar Nilai Mahasiswa
=====
Nama      Mata Kuliah 1Mata Kuliah 2Mata Kuliah 3Rata-Rata Keterangan
-----
Kahitna   75.5      85       90       83.50
Lanara    89.00     95.10    92.30    92.13
Candelaria 98.00     87.00    90.00    91.67
Lamie     78.00     90.00    97.00    88.33
Reverie   91.00     97.00    100.00   96.00      Terbaik
=====
Mahasiswa dengan rata-rata tertinggi: Reverie (96.00)

```

c. Deskripsi Program

Program ini digunakan untuk menyimpan dan menampilkan nilai 3 mata kuliah dari 5 mahasiswa. Program memanfaatkan konsep array dua dimensi, perulangan (loop), serta fungsi manipulasi output (setw dan setprecision) untuk menampilkan data dalam bentuk tabel yang rapi.

- 1) Library <iostream> digunakan untuk operasi input-output seperti cin dan cout.
- 2) Library <iomanip> digunakan untuk memformat tampilan output agar sejajar (fungsi setw, setprecision, dan fixed).
- 3) using namespace std;
Supaya tidak perlu menulis awalan std:: setiap kali menggunakan fungsi seperti cout atau cin.
- 4) Deklarasi Konstanta
const int MAHASISWA = 5;
const int MATAKULIAH = 3;
Menentukan jumlah mahasiswa (5) dan jumlah mata kuliah (3) yang tetap selama program berjalan.

5) Deklarasi Array dan Variabel

```
string nama[MAHASISWA];  
float nilai[MAHASISWA][MATAKULIAH];  
float rata[MAHASISWA];  
int terbaik = 0;
```

- nama[] untuk menyimpan nama mahasiswa.
- nilai[][] untuk menyimpan nilai dari 3 mata kuliah tiap mahasiswa.
- rata[] untuk menyimpan hasil rata-rata nilai tiap mahasiswa.
- terbaik menyimpan posisi mahasiswa dengan rata-rata tertinggi.

6) Proses Input Data

```
for (int i = 0; i < MAHASISWA; i++) {  
    cin >> nama[i];  
    float total = 0;  
    for (int j = 0; j < MATAKULIAH; j++) {  
        cin >> nilai[i][j];  
        total += nilai[i][j];  
    }  
    rata[i] = total / MATAKULIAH;  
}
```

- Loop i mengatur mahasiswa ke-1 hingga ke-5.
- Loop j mengatur input nilai 3 mata kuliah.
- Nilai total dibagi 3 untuk menghitung rata-rata setiap mahasiswa.

7) Penentuan Mahasiswa Terbaik

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

Untuk membandingkan setiap rata-rata mahasiswa dan simpan indeks mahasiswa dengan nilai tertinggi.

8) Penampilan Tabel

```
cout << left << setw(12) << "Nama"  
    << setw(10) << "Mata Kuliah 1"  
    << setw(10) << "Mata Kuliah 2"  
    << setw(10) << "Mata Kuliah 3"  
    << setw(10) << "Rata-Rata"  
    << "Keterangan\n";
```

- setw() mengatur lebar kolom.
- setprecision(2) menampilkan angka dua digit di belakang koma.
- Mahasiswa dengan rata-rata tertinggi diberi tanda “Terbaik” di kolom keterangan.

9) Menampilkan Hasil Akhir

```
cout << "Mahasiswa dengan rata-rata tertinggi: "
```

```
<< nama[terbaik] << " (" << rata[terbaik] << ")";
```

Menampilkan nama dan nilai rata-rata mahasiswa terbaik.

Program ini menunjukkan penerapan array dua dimensi untuk menyimpan data numerik, disertai perhitungan rata-rata dan pemilihan nilai maksimum menggunakan struktur perulangan.

Hasil akhir berupa tabel nilai mahasiswa yang tersusun rapi, dengan penandaan khusus untuk mahasiswa yang memiliki nilai rata-rata tertinggi.

2. Unguide 2

a. Source Code

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

void hitungPointer(int *ptr, int N, int &jumlah, int &maks, int &min);

int main() {
    int N;
    cout << "Masukkan jumlah elemen array: ";
    cin >> N;

    int *ptr = new int[N];

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

    int jumlah, maks, min;

    hitungPointer(ptr, N, jumlah, maks, min);

    cout << "\nHasil Perhitungan:\n";
    cout << "Jumlah seluruh bilangan : " << jumlah << endl;
    cout << "Nilai maksimum          : " << maks << endl;
    cout << "Nilai minimum            : " << min << endl;

    delete[] ptr;
    return 0;
}

void hitungPointer(int *ptr, int N, int &jumlah, int &maks, int &min) {
    jumlah = 0;
    maks = *ptr;
    min = *ptr;
```

```

    for (int i = 0; i < N; i++) {
        jumlah += *(ptr + i);
        if (*(ptr + i) > maks)
            maks = *(ptr + i);
        if (*(ptr + i) < min)
            min = *(ptr + i);
    }
}

```

b. Screenshot Output

```

Masukkan jumlah elemen array: 5

Masukkan 5 bilangan bulat:
Elemen ke-1 : 19
Elemen ke-2 : 17
Elemen ke-3 : 20
Elemen ke-4 : 90
Elemen ke-5 : 85

Hasil Perhitungan:
Jumlah seluruh bilangan : 231
Nilai maksimum          : 90
Nilai minimum           : 17

```

c. Deskripsi Program

Program ini digunakan untuk menghitung jumlah total, nilai maksimum, dan nilai minimum dari sekumpulan bilangan bulat yang menggunakan pointer untuk mengakses dan memproses elemen-elemen array.

1) Deklarasi dan Input Data

- Program meminta pengguna untuk memasukkan banyaknya elemen array (N).
- Kemudian program mengalokasikan memori dinamis untuk array dengan ukuran N menggunakan:
`int *ptr = new int[N];`
- Setelah itu, pengguna diminta mengisi setiap elemen array.

2) Pemanggilan Fungsi

Setelah semua diinput, program akan memanggil fungsi:

`hitungPointer(ptr, N, jumlah, maks, min);`

3) Proses di Fungsi `hitungPointer()`

- Penginisialisasian nilai awal:
`jumlah = 0;`
`maks = *ptr;`
`min = *ptr;`
- Lalu melakukan perulangan dari `i = 0` sampai `i < N`, dan di setiap iterasi: menambah nilai elemen ke jumlah, membandingkan untuk mencari nilai terbesar dan terkecil, semua elemen diakses menggunakan pointer aritmatika: `*(ptr + i)`.

4) Menampilkan Hasil

Setelah fungsi selesai, hasil jumlah, maks, dan min ditampilkan di fungsi main.

3. Unguide 3

a. Source Code

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

float hitungRata(float nilai[], int N);
void cariNilai(float nilai[], int N, float &maks, float &min);

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

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

    rata = hitungRata(nilai, N);
    cariNilai(nilai, N, maks, min);

    cout << "\nHasil Perhitungan:\n";
    cout << "Rata-rata kelas : " << rata << endl;
    cout << "Nilai tertinggi : " << maks << endl;
    cout << "Nilai terendah : " << min << endl;

    return 0;
}

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

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

```

        min = nilai[i];
    }
}

```

b. Screenshot Output

```

Masukkan jumlah siswa: 3
Nilai siswa ke-1 : 90
Nilai siswa ke-2 : 100
Nilai siswa ke-3 : 60

Hasil Perhitungan:
Rata-rata kelas : 83.3333
Nilai tertinggi : 100
Nilai terendah : 60
PS C:\Users\ACER\OneDrive\Desktop\L-Minggu 2>

```

c. Deskripsi Program

Program tersebut digunakan untuk menghitung rata-rata kelas, serta menentukan nilai tertinggi dan terendah dari sejumlah siswa yang nilainya diinput oleh pengguna. Terdapat dua fungsi utama:

- 1) `hitungRata()` → menghitung rata-rata seluruh nilai siswa.
- 2) `cariNilai()` → menentukan nilai tertinggi dan nilai terendah.

Program ini diawali dengan meminta pengguna memasukkan jumlah siswa (N), kemudian pengguna menginput nilai ujian untuk setiap siswa yang disimpan dalam array satu dimensi `nilai[N]`. Setelah semua data dimasukkan, fungsi `hitungRata()` dipanggil untuk menghitung rata-rata kelas dengan cara menjumlahkan seluruh elemen array dan membaginya dengan jumlah siswa (N). Selanjutnya, fungsi `cariNilai()` digunakan untuk menentukan nilai tertinggi dan nilai terendah dengan memanfaatkan parameter referensi (`&maks` dan `&min`). Setelah kedua fungsi selesai dijalankan, program menampilkan hasil perhitungan berupa nilai rata-rata kelas, nilai tertinggi, dan nilai terendah ke layar.

4. Unguide

a. Source Code

```

#include <iostream>
using namespace std;

void cetakSegitiga(int n);

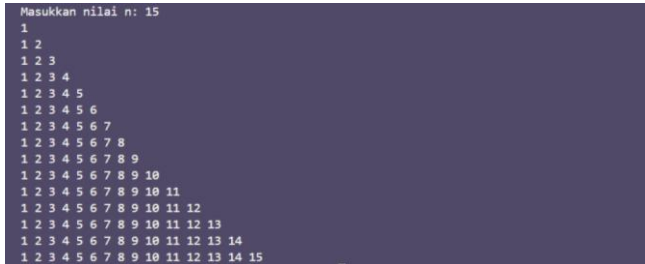
int main() {
    int n;
    cout << "Masukkan nilai n: ";
    cin >> n;

    cetakSegitiga(n);
    return 0;
}

```

```
void cetakSegitiga(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
}
```

b. Screenshot Output



c. Deskripsi Program

Program di atas digunakan untuk menampilkan pola segitiga angka berdasarkan bilangan bulat `n` yang dimasukkan oleh pengguna. Setelah pengguna memasukkan nilai `n`, program akan memanggil prosedur `cetakSegitiga(n)` yang bertugas mencetak pola angka dengan menggunakan dua perulangan. Perulangan luar (`i`) mengatur jumlah baris segitiga, sedangkan perulangan dalam (`j`) mencetak angka dari 1 hingga `i` pada setiap barisnya. Setelah satu baris selesai, program berpindah ke baris berikutnya menggunakan `endl`. Hasil akhirnya adalah tampilan pola segitiga angka yang meningkat secara bertahap dari baris pertama hingga baris ke-`n`.

D. Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa pengenalan bahasa C++ memberikan pemahaman dasar tentang penggunaan variabel, pointer, dan array, serta penerapan perulangan dan fungsi dalam pengolahan data. Pointer digunakan untuk mengakses dan memanipulasi nilai melalui alamat memori, sedangkan array berfungsi menyimpan data secara terstruktur. Melalui serangkaian latihan, kegiatan ini menunjukkan pentingnya logika dan penulisan kode yang sistematis dalam pemrograman. Secara keseluruhan, praktikum ini menjadi dasar penting untuk memahami konsep-konsep lanjutan pada materi struktur data dan algoritma.

E. Referensi

Deitel, H. M., & Deitel, P. J. (2012). C++ How to Program (8th ed.). Pearson Education.

Stroustrup, B. (2013). The C++ programming language (4th ed.). Addison-Wesley.

Tutorialspoint. (2024). C++ functions. Retrieved from https://www.tutorialspoint.com/cplusplus/cpp_functions.htm

Wicaksono, A. (2020). Belajar dasar pemrograman C++ untuk pemula. Deepublish.