

Tugas Kelompok Pemrograman Terstruktur Pemrograman C++



Disusun oleh:

- Adam Kurnia (201710225262)
- Fahrizal Cipto R. (201710225267)
- Roberto Lamas S. (201710225178)
- Tegar San Yofa (201710225288)

FAKULTAS TEKNIK INFORMATIKA
UNIVERSITAS BHAYANGKARA
JAKARTA RAYA

C++ adalah bahasa pemrograman komputer yang di buat oleh Bjarne Stroustrup, yang merupakan perkembangan dari bahasa C dikembangkan di Bell Labs (Dennis Ritchie) pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa sebelumnya, yaitu B, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (American National Standard Institute) Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded, Bjarne Stroustrup pada Bell Labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming).

SEJARAH BAHASA PEMROGRAMAN C++

Bahasa C++ diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories awal tahun 1980-an berdasarkan C ANSI (American National Standard Institute). Pertama kali, prototype C++ muncul sebagai C yang diperkanggih dengan fasilitas kelas. Bahasa tersebut disebut C dengan kelas (C with class). Selama tahun 1983-1984, C dengan kelas disempurnakan dengan menambahkan fasilitas pembebanan lebih operator dan fungsi yang kemudian melahirkan apa yang disebut C++. Symbol ++ merupakan operator C untuk operasi penarikan, muncul untuk menunjukkan bahwa bahasa baru ini merupakan versi yang lebih canggih dari C. Borland International merilis compiler Borland C++ dan Turbo C++. Kedua compiler ini sama-sama dapat digunakan untuk mengkompilasi kode C++. Bedanya, Borland C++ selain dapat digunakan dibawah lingkungan DOS, juga dapat digunakan untuk pemrograman Windows.

PENGERTIAN BAHASA PEMROGRAMAN C++

C++ merupakan bahasa pemrograman yang memiliki sifat Pemrograman berorientasi objek, Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan menjelaskan class-class yang merupakan anak class yang dibuat sebelumnya sebagai abstraksi dari object-object fisik, Class tersebut berisi keadaan object, anggota-anggotanya dan kemampuan dari objectnya, Setelah beberapa Class dibuat kemudian masalah dipecahkan dengan Class. Bahasa C adalah bahasa pemrograman prosedural yang memungkinkan kita untuk membuat prosedur dalam menyelesaikan suatu masalah. Bahasa pemrograman C++ adalah bahasa pemrograman yang berorientasi pada objek. Perbedaan Antara Bahasa pemrograman C dan C++ meskipun bahasa-bahasa tersebut menggunakan sintaks yang sama tetapi mereka memiliki perbedaan, C merupakan bahasa pemrograman prosedural, dimana penyelesaian suatu masalah dilakukan dengan membagi-bagi masalah tersebut kedalam sub-masalah yang lebih kecil, sedangkan untuk C++ merupakan bahasa pemrograman yang memiliki sifat Pemrograman berorientasi objek, Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan menjelaskan class-class yang merupakan anak class yang dibuat sebelumnya sebagai abstraksi dari object-object fisik, Class tersebut berisi keadaan object, anggota-anggotanya dan kemampuan dari objectnya, Setelah beberapa Class dibuat kemudian masalah dipecahkan dengan Class.

STANDARISASI

| Year | C++ Standard | Informal name |
|------|--------------------------------------|---------------|
| 2011 | ISO/IEC 14882:2011 ^[4] | C++11 |
| 2007 | ISO/IEC TR 19768:2007 ^[5] | C++TR1 |
| 2003 | ISO/IEC 14882:2003 ^[6] | C++03 |
| 1998 | ISO/IEC 14882:1998 ^[7] | C++98 |

Pada tahun 1998, C++ Standar Komite (ISO/IEC JTC1/SC22/WG21 Working Group) mengeluarkan Standar Internasional ISO/IEC 14882:1998 yang digunakan selama beberapa tahun. Pada tahun 2003 dirilis versi yang telah dikoreksi, ISO/IEC 14882:2003. Pada tahun 2005, Laporan Teknis yang disebut "Laporan Teknis Perpustakaan 1" (sering dikenal sebagai TR1), dirilis. Revisi terbaru dari standar C++ adalah C++11 (sebelumnya dikenal C++0x) telah disetujui oleh ISO/IEC pada tanggal 12 Agustus 2011.

CONTOH PROGRAM C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "hello world" << endl;
}
```

KETERANGAN

Baris pertama :

```
#include <iostream.h>
```

Sebagai bagian dari proses kompilator, Kompilator dari c++ menjalankan program yang dinamakan preprosesor. Preprosesor memiliki kemampuan menambahkan dan menghapus kode dari sumber, Pada bagian #include memberitahuakan preprosesor untuk menyertakan kode dari iostream, Berkas iostream berisi deklarasi untuk berbagai fungsi yang dibutuhkan oleh perangkat lunak, atau class-class yang dibutuhkan.

Baris kedua :

```
int main ()
```

Pernyataan ini mendeklarasikan fungsi utama, bahwa suatu program C++ dapat berisi banyak fungsi, yang harus selalu memiliki sebuah fungsi utama (main function), Fungsi adalah modul yang berisi kode-kode untuk menyelesaikan masalah-masalah tertentu. Kata Void menandakan fungsi main tidak bertipe.

Baris ketiga :

```
{
```

Kurung kurawal buka menandakan awal program.

Baris keempat :

```
std::cout << "Hello world" << endl;
```

Cout adalah sebuah object dari Pustaka perangkat lunak standart C++ yang digunakan untuk mencetak string ke piranti output standart, yang biasanya adalah layar komputer, Compiler menghubungkan kode dari pustaka perangkat lunak standar itu dengan kode yang telah ditulis untuk mendapatkan hasil executable.

Baris kelima:

```
}
```

Kurung kurawal tutup menandakan akhir program.

KATA YANG DIPESAN

KELOMPOK PERTAMA

C++ mempunyai 32 buah kata yang dipesan (*reserved words*), Kata kunci kelompok pertama merupakan turunan dari bahasa C, di antaranya:

| | | | | | | | |
|-------|----------|--------|-------|----------|--------|---------|----------|
| auto | const | double | float | int | Short | struct | unsigned |
| break | continue | Else | for | long | Signed | switch | Void |
| case | default | enum | goto | register | Sizeof | typedef | volatile |
| char | do | extern | if | return | Static | union | While |

KELOMPOK KEDUA

Kata yang dipesan kelompok kedua berjumlah 30. Kata-kata ini adalah baru dan hanya ada di bahasa C++:

| | | | | |
|------------|--------------|-----------|------------------|----------|
| asm | dynamic_cast | namespace | reinterpret_cast | Try |
| bool | explicit | new | static_cast | Typeid |
| catch | false | operator | template | typename |
| class | friend | private | This | Using |
| const_cast | inline | public | Throw | Virtual |
| delete | mutable | protected | True | wchar_t |

Kata-kata yang dipesan tersebut di atas tidak boleh dipakai sebagai nama variable, class, enum, macro, dan struct.

#include <stdio.h>

```
int maksimum (int,int); void tulis (int);
```

```
main() int nilai 1,nilai 2,nilai maks; printf (*\n ketikkan dua bilangan bulat:); scanf("%d%d",&nilai 1,nilai 2);
```

```
nilai maks=maksimum (nilai 1,nilai 2); tulis(nilai maks); returns 0;
```

```
int maksimum(int x1,int x2) if (x1>x2) return x1; else return x2;
```

```
void tulis(int x) printf("\n hasilnya adalah :%d",x);
```

TIPE DATA DASAR

Untuk menyimpan suatu variabel diperlukan tempat khusus di dalam memori komputer, Besar dan tipe dari variabel-variabel di dalam standar program C++ dispesifikasikan sebagai berikut.

| Nama | Keterangan | Ukuran | Jangkauan |
|-------------------|--|--------|--|
| Char | Abjad/karakter atau untuk bilangan bulat kecil | 1 byte | signed: -128 to 127 unsigned: 0 to 255 |
| Short int (short) | Bilangan bulat dengan jangkauan pendek | 2 byte | signed: -32768 to 32767 unsigned: 0 to 65535 |
| Int | Bilangan bulat | 4 byte | signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295 |
| long int | Integer dengan jangkauan panjang | 4 byte | signed: -2147483648 to |

| | | | |
|-------------|--|--------|---|
| (long) | | | 2147483647 unsigned: 0 to 4294967295 |
| Bool | Boolean, dapat bernilai benar atau salah (true or false) | 1 byte | true or false |
| Float | Angka dengan titik mengambang (bilangan cacah) | 4 byte | 3.4e +/- 38 (7 digit) |
| Double | Bilangan cacah dengan ketelitian ganda | 8 byte | 1.7e +/- 308 (15 digits) |
| long double | Bilangan cacah dengan ketelitian ganda panjang | 8 byte | 1.7e +/- 308 (15 digits) |
| wchar_t | Karakter lebar, biasa dipakai untuk Unicode karakter | 2 byte | 1 karakter lebar |

DASAR INPUT & OUTPUT (I/O) PADA C++ DENGAN FUNGSI CIN DAN COUT

#1 OUTPUT PADA C++ (SINTAKS: COUT)

Dalam pemrograman c++ kita akan sering menggunakan sintaks cout untuk menampilkan data ke alat output khususnya data berupa teks. Untuk menggunakan keyword cout kita membutuhkan Insertion Operator: yaitu 2 buah tanda lebih kecil << di antara keyword dengan ekspresi. Berikut contoh programnya:

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

int main()
{
    string selamat="Welcome to C++"
    cout << selamat;

    return 0;
}
```

Pada contoh kode diatas kita mendeklarasikan sebuah variabel bernama 'selamat' dengan tipe data string. Kemudian kita mencetaknya dengan keyword cout. Anda hanya perlu menuliskan variabelnya setelah Insertion Operator. Namun jika kita ingin langsung menampilkan sebaris kalimat (string), kita dapat menambahkan sepasang tanda petik dua di awal kalimat dan di akhir kalimat. Berikut contoh programnya:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Welcome to C++";

    return 0;
}
```

Seperti yang anda lihat kita hanya perlu membungkus teks didalam dua tanda kutip.

#2 INPUT PADA C++ (SINTAKS: CIN)

Jika fungsi cout kita gunakan untuk menampilkan output maka bisa ditebak bahwa fungsi cin akan kita gunakan untuk melakukan perintah input dasar text. Untuk menggunakan keyword cin kita membutuhkan Extraction Operator dengan 2 buah tanda lebih dari >> yang diletakan di antara keyword cin dan memori. Untuk itu, kita harus mendeklarasikan variabel terlebih dahulu. Karena nantinya, data yang diberikan oleh pengguna dalam operasi pemasukan (input) akan disimpan di dalam memori variable yang telah kita deklarasikan. Silahan perhatikan baris kode berikut:

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

string nama;

int main()
{
    cout << "Masukan Nama: ";
    cin >> nama;

    cout << "Nama saya adalah " << nama;
}
```

Dari contoh program diatas, kita telah mendeklarasikan variabel nama bertipe string. Sedangkan cin untuk meyimpan nilai masukan dari pengguna ke variabel nama. Lalu nilai/value dari variabel nama ditampilkan dengan menggunakan fungsi cout.

MENARIK KESIMPULAN

Baiklah akan saya rangkum kembali untuk kesimpulan, dalam program c++ kita akan mengenal dua fungsi bernama cin untuk input teks dan cout untuk output teks. Untuk menggunakannya, kita mebutuhkan sebuah tambahan operator:

Operator Insertion << (2 buah tanda kurang dari) untuk cout.

Operator Extraction >> (2 buah tanda lebih dari) untuk cin.

Khusus untuk cout anda juga bisa langsung mecetak teks string yang dibungkus dengan double quotes tanpa harus menggunakan variabel tambahan.

CONTOH PROGRAM C++ PERULANGAN SEDERHANA DAN PENJELASANNYA

Pengertian Perulangan atau Looping/Perulangan dapat diartikan sebagai berikut:

Perulangan (atau dalam bahasa inggris disebut dengan loop) adalah instruksi program yang bertujuan untuk mengulang beberapa baris perintah. Fungsi paling sederhana yaitu untuk mempermudah melakukan suatu proses yang berulang-ulang, seperti mencetak angka dari 1 – 100. Ada 3 cara untuk melakukan perulangan di C++ yaitu: for, while, & do while.

#1 PERULANGAN DENGAN FUNGSI FOR

Penulisan dasar format perulangan for dalam C++ adalah sebagai berikut:

```
for (start, condition; increment)
{
    statement;
}
```

Start adalah kondisi pada saat awal perulangan. Biasanya kondisi awal ini digunakan untuk membuat dan memberikan nilai kepada variabel yang digunakan untuk mengontrol perulangan. Misalkan, kita akan membuat variabel *i*, maka untuk kondisi start ini, kita juga harus memberikan nilai awal untuk variabel *i*, misalnya dengan 1, maka *i*=1. Misalkan kita ingin menghentikan perulangan jika variabel *i* telah mencapai nilai 10, maka pada bagian condition ini kita membuat perintah *i*≤10. Yang berarti selama nilai *i* kurang atau sama dengan 10, terus lakukan perulangan.

Increment adalah bagian yang digunakan untuk memproses variabel agar bisa memenuhi kondisi akhir perulangan. Umumnya nilai variable tersebut bertambah (*i*++) / berkurang (*i*--) 1 (satu).

Condition adalah kondisi yang harus dipenuhi agar perulangan dijalankan. Selama kondisi ini terpenuhi, maka C++ akan terus melakukan perulangan.

Statement adalah bagian kode program yang akan diproses secara terus-menerus selama proses perulangan berlangsung. Kita membuat blok program di antara tanda kurung kurawal ({dan}) sebagai penanda bahwa bagian di dalam kurung kurawal inilah yang akan dikenai proses perulangan. Contoh looping dengan fungsi for pada program C++:

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

void CetakAngka (int i, int j){
    for (i = 1; i <= j; i++){
        cout << i << "\n";
    }
}

int main ()
{
    CetakAngka (1,10);
}
```

#2 PERULANGAN DENGAN FUNGSI WHILE

Dalam perulangan while, program akan terus melakukan perulangan dengan mengeksekusi pernyataan target selama kondisi tertentu bernilai benar. Penulisan dasar format perulangan while dalam C++ adalah sebagai berikut:

```
while(condition){
    statement;
}
```

Condition adalah kondisi yang harus dipenuhi agar perulangan berlangsung. Kondisi ini mirip seperti dalam perulangan for. **Condition** ini akan diperiksa pada tiap perulangan, dan hanya jika hasilnya FALSE, maka proses perulangan berhenti. Artinya kita tidak tahu berapa banyaknya perulangan. Karena, selama condition bernilai TRUE, maka perulangan akan terus dilakukan. **Statement** adalah kode program yang akan diulang. **Tanda kurung kurawal** diperlukan untuk

membatasi blok program yang akan diulang. Jika statement hanya terdiri dari 1 baris, maka tidak diperlukan. Contoh perulangan dengan fungsi while dalam program C++:

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

int main () {
    // Local variable declaration:
    int a = 1;

    // while loop execution
    while( a < 10 ) {
        cout << "value of a: " << a << endl;
        a++;
    }

    return 0;
}
```

#3 PERULANGAN DENGAN FUNGSI DO WHILE

Perulangan while dan do-while pada dasarnya hampir sama.

```
do {
    statement;
} while (condition);
```

Perbedaan terletak pada 'lokasi' pengecekan kondisi perulangan. Dalam struktur while, pengecekan untuk kondisi perulangan di lakukan di awal, sehingga jika kondisi tidak terpenuhi, maka perulangan tidak akan pernah dijalankan. Namun pada perulangan do-while: Pengecekan kondisi akan dilakukan di akhir perulangan, sehingga walaupun kondisi adalah FALSE, perulangan akan tetap berjalan minimal 1 kali. **Statement** adalah kode program yang akan diulang & **condition** adalah kondisi yang harus dipenuhi agar perulangan berlangsung. Contoh program looping dengan fungsi do while dalam pemrograman C++:

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

int main () {
    // Local variable declaration:
    int a = 1;

    // while loop execution
    do {
        cout << "value of a: " << a << endl;
        a = a + 1;
    } while ( a < 10 );

    return 0;
}
```

MENARIK KESIMPULAN

Meskipun perulangan atau looping memiliki fungsi yang sama, namun ketiga fungsi tersebut digunakan pada kondisi yang berbeda. While - Perulangan while digunakan untuk kondisi perulangan dimana banyaknya perulangan tidak dapat dipastikan pada saat penulisan program. For - Perulangan for digunakan untuk kondisi perulangan dimana jumlah perulangan telah ditentukan di awal. do-while sama seperti perulangan while, dalam perancangan perulangan do-while ini kita harus memahami alur logika program yang dibuat. Banyaknya perulangan ditentukan pada saat program mencapai kondisi FALSE. Kesalahan dalam alur logika akan membuat C++ tidak pernah berhenti (infinity loop). Demikian penjelasan mengenai fungsi-fungsi perulangan atau looping pada bahasa pemrograman C++.

KELEBIHAN DARI BAHASA PEMROGRAMAN C++

Bahasa C++ tersedia hampir di semua jenis computer. Kecepatan program jika dibanding dengan program yang sama buatan bahasa lain, C++ relatif lebih cepat. C++ adalah bahasa yang terstruktur, dengan demikian akan lebih mendukung OOP. Bahasa Standard-nya ANSI bisa dipakai diberbagai platform.

KEKURANGAN DARI BAHASA PEMROGRAMAN C++

Sulitnya untuk membuat sesuatu dengan Bahasa Pemrograman C++ sehingga minat dalam memperdalam programming akhirnya harus terhenti, kecuali dengan inisiatif sendiri mempelajari bahasa/teknologi lain. C++ tidak murni OOP sehingga kurang cocok untuk mengajarkan Konsep OOP karena kaidah-kaidah OOP dapat dilanggar. Di dalam bahasa pemrograman C++ terdapat Pengarah Preprocessor. Dalam Bahasa Pemrograman C atau C++ bagian pertama yang menyusun sebuah tugas, terbentuk di pengarah preprocessor. Pengarah preprocessor menginstruksikan compiler untuk membentuk beberapa tugas sebelum kompilasi dimulai. Pengarah preprocessor `#include`, salah satu yang sering terlihat, contoh `#include<stdio.h>` menginstruksikan compiler untuk menyertakan berkas C++ sumber yang lain sebelum kompilasi dimulai. Pengarah preprocessor yang lain adalah `#define`, yang merupakan definisi sebuah macro.

PROGRAM YANG DIBUAT

Program ini dibuat berfungsi untuk menguraikan sebuah kalimat dipecah menjadi karakter-karakter, program ini dibuat dengan tujuan agar kita mengetahui karakter apa saja yang sudah kita ketik dan berapa jumlah karakter yang kita ketik.

```
#include <iostream>
#include <conio.h>
#include <string>
#include <stdio.h>
#include <stdlib.h>
using namespace std;

string getVocal(std::string inputKalimat){
    char vocalPembanding[] = {'a','i','u','e','o','A','I','U','E','O'};
    string vocals;
    for(int i=0;i<inputKalimat.length();i++){
        for(int v=0;v<sizeof(vocalPembanding);v++){
            if(inputKalimat.at(i) == vocalPembanding[v]){
                vocals.push_back(vocalPembanding[v]);
            }
        }
    }
    return vocals;
}

string getKonsonan(std::string inputKalimat){
    char konsonanPembanding[] = {'b','c','d','f','g','h','j','k','l','m','n','p','q','r','s','t','v','w','x','y','z','B','C','D','F','G','H','J','K','L','M','N','P','Q','R','S','T','V','W','X','Y','Z'};
    string konsonans;
    for(int i=0;i<inputKalimat.length();i++){
        for(int k=0;k<sizeof(konsonanPembanding);k++){
            if(inputKalimat.at(i) == konsonanPembanding[k]){
                konsonans.push_back(konsonanPembanding[k]);
            }
        }
    }
    return konsonans;
}

string getAngka(std::string inputKalimat){
    int angkaPembanding [] = {'0','1','2','3','4','5','6','7','8','9'};
    string angka;
    for(int i=0;i<inputKalimat.length();i++){
        for(int a=0;a<sizeof(angkaPembanding);a++){
            if(inputKalimat.at(i) == angkaPembanding[a]){
                angka.push_back(angkaPembanding[a]);
            }
        }
    }
    return angka;
}

string getSpace(std::string inputKalimat){
    int spacePembanding [] = {' '};
```

```

        string spaces;
        for(int i=0;i<inputKalimat.length();i++){
            for(int s=0;s<sizeof(spacePembanding);s++){
                if(inputKalimat.at(i) == spacePembanding[s]){
                    spaces.push_back(spacePembanding[s]);
                }
            }
        }
        return spaces;
    }

string getUnique(std::string response){
    string duplicates;
    for(int i=0;i<response.length();i++){
        char b = response.at(i);
        if(i==0 || response.at(i) != response.at(i-1)){
            duplicates.push_back(response.at(i));
        }
    }
    return duplicates;
}

int calcTotal(std::string inputKalimat){
    return inputKalimat.length();
}

main(){
    char inputKalimat[500];
    string konsonan;
    string vocal;
    string angka;
    string space;
    string uniqueVocal;
    string uniqueKonsonan;
    string uniqueAngka;
    int totalChar;
    int totalVocal;
    int totalKonsonan;
    int totalAngka;
    int totalSpace;
    char lanjutkan;

    Pertanyaan:
        cout <<"Masukkan Kalimat Yang Ingin Diproses: ";
        cin.get(inputKalimat, 500);

        totalChar = calcTotal(inputKalimat);

        vocal = getVocal(inputKalimat);
        uniqueVocal = getUnique(vocal);
        totalVocal = calcTotal(vocal);

        konsonan = getKonsonan(inputKalimat);
        uniqueKonsonan = getUnique(konsonan);
        totalKonsonan = calcTotal(konsonan);

```

```

angka = getAngka(inputKalimat);
uniqueAngka = getUnique(angka);
totalAngka = calcTotal(angka);

space = getSpace(inputKalimat);
totalSpace = calcTotal(space);

cout <<"Total Character Dari Kalimat Yang Anda Masukkan: "<<totalChar<<endl<<endl;

cout <<"Character Vocal: "<<vocal<<endl;
cout <<"Total Character Vocal: "<<totalVocal<<endl<<endl;

cout <<"Character Konsonan: "<<konsonan<<endl;
cout <<"Total Character Konsonan: "<<totalKonsonan<<endl<<endl;

cout <<"Angka: "<<uniqueAngka<<endl;
cout <<"Total Angkan: "<<totalAngka<<endl<<endl;

cout <<"Total Space: "<<totalSpace<<endl<<endl;

```

Selesai:

```

cout <<"Lanjut? [Y/T] ";
cin >> lanjutkan;

if (lanjutkan == 'Y' || lanjutkan == 'y'){
    cin.clear();
    cin.ignore(INT_MAX, '\n');
    goto Pertanyaan;
    return 0;
}

```

}