# RANGKUMAN TEORI 1

## PENDAHULUAN

Saat melakukan pemrograman dalam bahasa pemrograman apa pun, Anda perlu menggunakan berbagai variabel untuk menyimpan berbagai informasi. Variabel hanyalah lokasi memori yang dicadangkan untuk menyimpan nilai. Ini berarti bahwa ketika Anda membuat variabel, Anda menyediakan beberapa ruang di memori.

Anda mungkin ingin menyimpan informasi dari berbagai tipe data seperti char, widechar, integer, floating point, double floating point, boolean dan lain-lain. Berdasarkan tipe data variabel, sistem operasi mengalokasikan memori dan memutuskan apa yang dapat disimpan di memori cadangan.

## TIPE DATA, VARIABEL DAN KONSTANTA

### Tipe Data

Tipe data adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Tipe data adalah jenis nilai yang akan disimpan pada suatu variabel.

1. Berikut adalah tipe data pada bahasa pemrograman C++:
2. **Void** (Kosong) adalah tipe data yang tidak bertipe karena mempunyai ukuran 0-byte biasanya digunakan untuk tipe data kosong seperti membuat fungsi.
3. **Char** (karakter) adalah tipe data untuk karakter yang sering digunakan untuk tipe data yang menggunakan huruf dan angka sebagai datanya.
4. **Float** (Floating Point) adalah tipe data untuk numerik yang digunakan untuk data berupa angka pecahan.
5. **Double** (Double Floating Point) adalah tipe data untuk numerik yang digunakan untuk data berupa angka pecahan yang nilai decimalnya dua kali lebih banyak.
6. **Bool** (Boolean) adalah tipe data yang digunakan untuk menentukan false dan true, tipe data ini sama seperti bilangan biner hanya ada dua angka saja yaitu 0 dan 1.
7. Modifikasi Tipe Data C++ beberapa macam tipe data dapat dimodifikasi sesuai kegunaannya dengan menambahkan fungsi ini didepan tipe data:  
   **Short** : Memodifikasi jangkauan nilai ke dalam bit yang lebih kecil   
    (16bit).

**Long** : Memodifikasi jangkauan nilai ke dalam bit yang lebih besar  
 (32bit).

**Signed** : Memodifikasi jangkauan nilainya negatif dan positif.

**Unsigned** : Memodifikasi jangkauan nilainya 0.

### Variabel

Secara umum variabel adalah sebuah nama lokasi penyimpanan di dalam memori, di mana Anda dapat menyimpan, mengubah dan mengambil nilai tersebut dengan memanggil nama variabel.

1. Pemberian Nama Variabel

Nama variabel ditentukan atau dikarang sendiri oleh pembuat program dengan syarat sebagai berikut:

1. Tidak boleh sama dengan nama atau kata yang sudah disiapkan oleh komputer (reserved word) seperti keyword, dan function. Juga harus berbeda dengan nama label dan konstanta yang dibuat oleh pemrogram.
2. Maksimum 32 karakter, bila lebih dari 32 karakter, maka karakter selebihnya tidak diperhatikan oleh komputer. Huruf besar dan huruf kecil berbeda.
3. Tidak boleh mengandung spasi atau blank.
4. Karakter pertama harus huruf atau karakter garis bawah (Under Score) dan karakter berikutnya boleh huruf atau angka atau karakter garis bawah.
5. Mendeklarasikan Variabel

Nama variabel dapat berupa kombinasi huruf apa saja, tetapi tidak boleh berisi spasi. Nama variabel legal termasuk x, J23qrsnf, dan myAge. Nama variabel yang baik memberitahu Anda untuk apa variabel tersebut. Menggunakan nama yang baik memudahkan untuk memahami alur program Anda. Pernyataan berikut mendefinisikan variabel integer yang disebut myAge:

int myAge;

Ingat bahwa C++ peka huruf besar/kecil (case sensitive), jadi myAge adalah nama variabel yang berbeda dari MyAge.

1. Membuat Lebih dari Satu Variabel Sekaligus

Anda dapat membuat lebih dari satu variabel berjenis sama dalam satu pernyataan dengan menulis tipe dan kemudian nama variabel, dipisahkan dengan koma. Sebagai contoh:

int myAge, myWeight; // dua variabel bertipe data int

long area, width, length; // tiga variabel bertipe data long

1. Menetapkan Nilai ke Variabel

Anda menetapkan nilai ke variabel dengan menggunakan operator (=). Berikut adalah contoh menetapkan nilai 20 ke myAge:

int myAge = 20;

atau

int myAge;

myAge = 20;

1. Sensitivitas Huruf pada Variabel

Bahasa pemrograman C++ peka terhadap huruf besar/kecil. Dengan kata lain, huruf besar dan kecil dianggap berbeda. Variabel bernama age berbeda dengan Age, dan berbeda dengan AGE.

### Konstanta

Seperti variabel, konstanta adalah lokasi penyimpanan data. Tetapi variabel dapat bervariasi, sedangkan konstanta tidak bervariasi. Sesuai dengan namanya, nilai dari konstanta bersifat tetap (konstan) dan tidak bisa diubah sepanjang program berjalan. Anda harus menginisialisasi sebuah konstanta saat Anda membuatnya, dan Anda tidak dapat menetapkan nilai baru nanti setelah sebuah konstanta diinisialisasi. C++ memiliki dua jenis konstanta: literal dan simbolis.

1. Konstanta Literal

Konstanta literal adalah nilai yang diketik langsung ke program Anda dimanapun diperlukan. Sebagai contoh:

int myAge = 39;

myAge adalah sebuah variabel, bertipe int, 39 adalah konstanta literal. Anda tidak dapat menetapkan nilai ke 39, dan nilainya tidak dapat diubah.

1. Konstanta Simbolis

Konstanta simbolis adalah konstanta yang diwakili oleh sebuah nama, seperti halnya variabel. Tidak seperti variabel, bagaimanapun, setelah konstanta diinisialisasi, nilainya tidak dapat diubah.

Jika program Anda memiliki satu variabel integer bernama students dan nama class lain, Anda dapat menghitung berapa banyak siswa yang Anda miliki, berdasarkan jumlah kelas yang diketahui, jika Anda mengetahui ada 15 siswa per kelas:

students = classes \* 15;

Dalam contoh ini, 15 adalah konstanta literal. Kode Anda akan lebih mudah dibaca dan lebih mudah dipertahankan jika Anda mengganti konstanta simbolis untuk nilai ini:

students = classes \* studentsPerClass;

Jika Anda memutuskan untuk mengubah jumlah siswa di setiap kelas, Anda dapat melakukannya di mana Anda menentukan studentsPerClass konstan tanpa harus membuat perubahan di setiap tempat Anda saat menggunakan nilai tersebut. Resikonya adalah Anda bisa melewatkan salah satu tempat di mana Anda menggunakan nilai dan mengalami kesulitan untuk men-debug kesalahan logika.

1. Mendeklarasikan Konstanta dengan #define

Untuk mendefinisikan cara yang kuno dan tidak benar, Anda akan memasukkan:

#define studentsPerClass 15;

Begitulah cara melakukannya di versi lama bahasa C, standar ANSI memperkenalkan kata kunci const ke bahasa tersebut. Perhatikan bahwa studentsPerClass tidak memiliki tipe data. #define melakukan substitusi teks sederhana. Setiap kali preprocessor melihat kata studentsPerClass, ia menempatkan 15 di teks.

1. Mendefinisikan Konstanta dengan const

Meskipun #define berfungsi, ada cara yang lebih baik, tidak terlalu menggemukkan, dan lebih menarik untuk menentukan konstanta di C++:

const unsigned short int studentsPerClass = 15;

Contoh ini juga mendeklarasikan konstanta simbolis bernama studentsPerClass, tapi kali ini studentsPerClass diketik sebagai unsigned short int. ini membutuhkan waktu lebih lama untuk mengetik, tetapi menawarkan beberapa keuntungan. Perbedaan terbesar adalah konstanta ini memiliki tipe, dan compiler dapat memaksakannya untuk digunakan sesuai dengan tipenya.

1. Konstanta Enumerated

Konstanta yang disebutkan membuat sekumpulan konstanta. Misalnya, Anda dapat mendeklarasikan COLOR sebagai enumeration; dan Anda dapat menentukan bahwa ada lima nilai untuk COLOR: RED, BLUE, GREEN, WHITE, and BLACK. Sintaks untuk konstanta yang disebutkan adalah dengan menulis kata kunci enum, diikuti dengan nama jenis, tanda kurung kurawal buka, masing-masing nilai dipisahkan oleh koma dan diakhiri tanda kurung tutup kurawal dan titik koma. Berikut contohnya:

enum COLOR {RED, BLUE, GREEN, WHITE, BLACK};

Pernyataan ini melakukan dua tugas:

1) Itu membuat COLOR menjadi nama enumerasi, yaitu tipe baru

2) Itu membuat RED menjadi konstanta simbolis dengan nilai 0, BLUE

konstanta simbolis dengan nilai 1, GREEN sebagai konstanta simbolis dengan nilai 2, dan seterusnya.

Setiap konstanta yang disebutkan memiliki nilai integer. Jika Anda tidak menentukan sebaliknya, konstanta pertama akan bernilai 0, dan sisanya akan dihitung dari sana. Namun, salah satu konstanta dapat diinisialisasi dengan nilai tertentu, dan konstanta yang tidak diinisialisasiakan dihitung ke atas dari yang sebelumnya. Jadi, jika Anda menulis:

enum Color {RED = 100, BLUE, GREEN = 500, WHITE, BLACK = 700};

maka RED akan memiliki nilai 100, BLUE nilai 101, GREEN nilai 500, WHITE nilai 501 dan BLACK nilai 700.

Keuntungan dari teknik ini adalah Anda bisa menggunakan nama simbolis seperti BLACK atau WHITE alih-alih angka yang mungkin tidak berarti seperti 1 atau 700.

Merupakan praktik umum untuk menggunakan semua huruf kapital untuk nama konstanta reguler dan enumerasi. Ini membuatnya jelas secara visual bahwa nama yang Anda baca adalah konstanta dan bukan variabel yang boleh Anda ubah. Compiler itu sendiri tidak peduli selama Anda konsisten.

# TUGAS PENDAHULUAN 1

1. Apa itu Variabel?

Variabel adalah sebuah nama lokasi penyimpanan di dalam memori, di mana Anda dapat menyimpan, mengubah dan mengambil nilai tersebut dengan memanggil nama variabel.

1. Apa perbedaan dari initialization dan defining untuk sebuah variabel?

Initialization adalah memberi nilai awal yang akan disimpan dalam suatu variabel.

Contoh:   
int myAge = 20;

Sedangkan Defining adalah mendeklarasikan variabel dengan mengalokasikan penyimpanan untuk variabel tersebut.

Contoh:

int myAge;

1. Apa perbedaan antara sebuah #define constant dan const?

#define constant adalah salah satu cara mendeklarasikan sebuah konstanta tanpa mempedulikan tipe datanya.

Sedangkan Const adalah salah satu cara mendeklarasikan sebuah konstanta yang menggunakan tipe data.

1. Apa yang dimaksud dengan konstanta literal dan simbolis?

Konstanta literal adalah nilai yang diketik langsung ke program.

Sedangkan Konstanta simbolis adalah konstanta yang diwakili oleh sebuah nama, seperti halnya variabel.

1. Apakah variabel DOG, dog, Dog, doG sama? Jelaskan!

Berbeda, karena sebuah variabel memiliki sensitivitas huruf di mana huruf besar dan kecil dianggap berbeda pada bahasa pemrograman C++.

DOG = seluruh huruf besar.

dog = seluruh huruf kecil.

Dog = diawali huruf besar lalu, diikuti huruf kecil.

doG = seluruh huruf kecil lalu, diakhiri huruf besar.