

Struktur Dasar Program C++

Bahan Kuliah SD3105 Algoritma dan Pemrograman

Sevi Nurafni

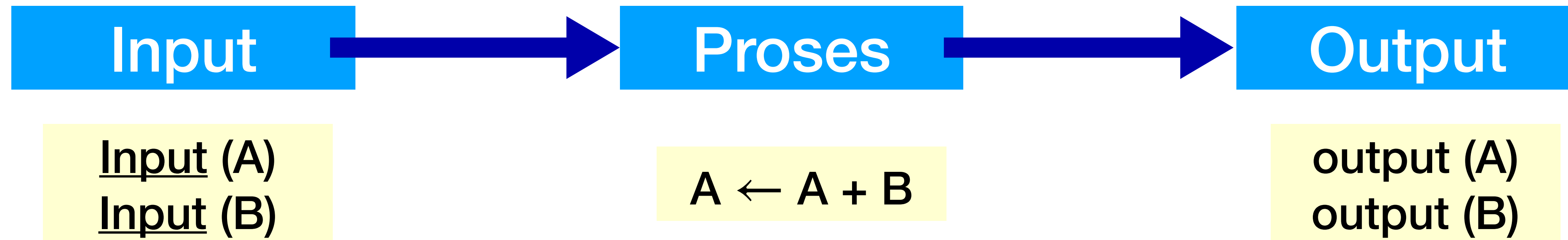
Fakultas Sains dan Teknologi

Universitas Koperasi Indonesia 2024

C++

- C++ merupakan bahasa pemrograman general purpose dan multi paradigma (prosedural, object oriented)
- Bahasa pemrograman yang sangat populer dan banyak digunakan
- Dikembangkan oleh Bjarne S. Stroustrup mulai tahun 1979 di Bell Labs
- Merupakan pengembangan dari Bahasa C (prosedural murni) dengan penambahan konsep, object-orientation
- Dalam kuliah ini, hanya akan menggunakan paradigma prosedural
- Merupakan bahasa yang **case sensitive** → perbedaan huruf besar dan kecil berpengaruh

Input - Proses - Output



C++

```
cin >> A;  
cin >> B;
```

```
A = A + B;
```

```
cout << A;  
cout << B;
```

Apa hasil eksekusinya?

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

int main () {
    //KAMUS
    int A; int B;

    //ALGORITMA
    cin >> A; cin >> B;

    A = A + B;

    cout << A << endl;
    cout << B << endl;
    return 0;
}
```

Apa hasil eksekusinya?

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

int main () {
    //KAMUS
    string nama;

    //ALGORITMA
    cout << "Tuliskan namamu: " << endl;

    cin >> nama ;

    cout << "Namamu adalah : " << nama << endl; return 0;

}
```

Apa hasil eksekusinya?

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

int main () {
    //KAMUS
    int A; int B;

    //ALGORITMA
    A = 10;
    B = 5;

    A = A + B;
    B = B - A;

    Cout << A << endl;
    Cout << B << endl;

    Return 0;
}
```

Header



- `iostream` adalah salah satu *header* file yang ada di C++. *Header* ini digunakan untuk fungsi input dan output yang ada di C++.
Contoh fungsi input/output: `cin` dan `cout`
- `using namespace std` adalah perintah yang digunakan untuk mendeklarasikan/memberitahukan kepada *compiler* C++ bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace `std`

Bagian Kamus



- Kamus dipakai untuk mendeklarasi nama-nama yang digunakan dalam program
- Deklarasi nama yang didefinisikan pemrogram
 - type
 - variabel
 - konstanta
- Deklarasi BUKAN instruksi
- Contoh deklarasi [variabel]:

C++

int l;

float JumlahUang;

Jenis-Jenis Tipe Data



- Tipe data primitif atau tipe dasar (dalam C++)
 - Boolean (bool)
 - Integer (int)
 - Real (float)
 - Character (char)
 - String (string)
- Tipe data turunan atau bentukan
 - Dibentuk dari gabungan tipe dasar
 - Contoh
 - Tipe Data Mahasiswa
 - Dibentuk dari
 - » NIM: string
 - » Nama: string
 - » Umur: integer
 - » Kota: string
 - Tipe Array
 - Dibentuk dari kumpulan integer, misalnya 10 data tentang umur

Contoh Tipe Data

Umur	→ Integer contoh: 25, 44, 35
Kota	→ String, contoh: "Jakarta", "Bandung"
Nama	→ String, contoh: "Budi", "Ali"
Suhu	→ Integer atau float, contoh: 37.5 , 100
Luas	→ Integer atau float, contoh: 400, 43.5
BeratBadan	→ Integer atau float, contoh: 60.5, 75

Contoh deklarasi tipe bentukan/komposit/struct

```
// Kamus
typedef struct {
    int x;
    int y;
} Point;
typedef struct {
    string NIM;
    string Nama;
    int Umur;
    string Kota;
} DataMahasiswa;
```

Variabel



- Variabel menyimpan nilai ber-"tipe data" sesuai dengan deklarasi
- Variabel :
 - deklarasi (supaya nama dikenal),
 - inisialisasi nilai (siap dimanipulasi)
- Contoh
 - Deklarasi variabel
`int i; float A;`
 - Inisialisasi variabel
`i = 100;`
 - Artinya variabel i di isi dengan nilai 100`A = 8.25;`
 - Artinya variabel A diisi dengan nilai real 8.25
- Operasi terhadap variabel sangat tergantung dari tipe datanya.

Operasi pada nilai suatu tipe data

- Operasi perhitungan akan memerlukan operator seperti “+”, “-”, “*” dan “/” (tambah, kurang, kali dan bagi) untuk melakukan kalkulasi
- Operasi “+” pada tipe data bukan numerik memiliki arti yang berbeda
 - Contoh: “Halo “ + “Apa kabar “ → “Halo Apa kabar
- Tidak semua operator dapat digunakan untuk tipe data numerik.
 - Contoh: “Halo “ * “Apa kabar “ → **X**

Operasi tipe dasar

- int: * / + - % < > <= >= == !=
- bool: && || ! !=
- float: * / + - < > <= >= !=
- char : == !=

Membuat Nama Variabel yang benar dan “baik”



- Nama variabel harus dimulai dengan huruf dan dapat diikuti dengan huruf lagi dan angka
 - Tidak boleh ada tanda baca kecuali underscore (_)
- Dalam nama variabel tidak boleh dipisahkan oleh spasi
- Cari nama variabel yang bisa dimengerti
 - Agar tidak membingungkan
- C++ adalah bahasa yang **case sensitive** → Kesalahan penulisan huruf besar dan kecil menyebabkan error

Konstanta



- Berbeda dengan Variable, suatu konstanta tidak boleh diubah nilainya
- Contoh
 - const float PI = 3.1415
 - const int nilai = 1000
- Pemakaian yang salah
 - PI = 44.5
 - nilai = 5000

Assignment



- Ruas kiri = Ruas Kanan ;
- Ruas kiri harus variable
- Ruas kanan harus <ekspresi>
- Ekspresi :
 - “rumus perhitungan”
 - Contoh:
 - $\text{Luas} = \text{panjang} * \text{lebar}$

Ekspresi

- Ekspresi Aritmatika

$$A + B$$

$$x + 2 * y$$

$$P - 2 * Q + R/S$$

- Ekspresi Relasional (pembandingan)

$$A < B \quad X == Y$$

$$\text{Total} \geq \text{nilai}$$

- Ekspresi Logika

$$A \ \&\& \ B \ C \ || \ B$$

Komentar

- Dalam bahasa pemrograman komentar adalah bagian program yang tidak dieksekusi
 - Bagian ini hanya digunakan untuk memberikan penjelasan suatu langkah, rumus ataupun bisa hanya berupa keterangan
- Dalam C++, komentar dituliskan sebagai:
 - Antara `/*` dan `*/`
`/* ini komentar */`
 - Diawali dengan `//`
`// ini komentar`

Aksi Sekuensial



- Aksi sekuensial
 - sederetan instruksi primitif dan/atau aksi yang akan dilaksanakan (dieksekusi) oleh komputer berdasarkan urutan penulisannya
 - Setiap aksi akan mengubah status dari program
 - Jadi setiap aksi sekuensial harus ada awal dan akhir.
 - atau dengan kata lain suatu program harus dimulai dan suatu ketika harus berakhir
 - Program yang tidak pernah berhenti adalah program yang salah atau error

Penulisan untuk instruksi Sekuensial



- Instruksi ditulis terurut sesuai penulisan per baris
- Setiap instruksi selalu diakhiri dengan tanda titik koma
 - Jadi dalam satu baris dapat terdiri dari lebih dari satu instruksi.

Contoh Aksi Sekuensial

/* contoh aksi sekuensial per baris */

```
int main(){  
    /* Kamus */  
    int i;  
    float x;  
  
    /* Algoritma */  
    cin >> i;  
    x = 100.75;  
  
    cout << x << endl;  
    cout << i * 2 << endl;  
  
    return 0;  
}
```

/* contoh aksi sekuensial dg titik koma */

```
int main(){  
    /* Kamus */  
    int i;  
    float x;  
  
    /* Algoritma */  
    cin >> i; x = 100.75;  
  
    cout << x << endl;   cout << i * 2 << endl;  
  
    return 0;  
}
```

Penulisan untuk instruksi Sekuensial



- Perhatikan bahwa:
 - ada program yang akan berubah jika urutan baris instruksinya berubah
 - dan ada juga program yang tidak berubah jika urutan baris instruksinya berubah

Pengubahan urutan eksekusi yang tidak merubah hasil eksekusi

```
/* contoh aksi sekuensial per  
baris */
```

```
int main()  
{  
    /* Kamus */  
    int i;  
    float x;
```

```
    /* Algoritma */
```

```
    cin >> i;  
    x = 100.75;
```

```
    cout << x << endl;  
    cout << i * 2 << endl;  
    return 0;
```

```
}
```

```
/* contoh aksi sekuensial per  
baris */
```

```
int main()  
{  
    /* Kamus */  
    float x;  
    int i;
```

```
    /* Algoritma */
```

```
    x = 100.75;  
    cin >> i;
```

```
    cout << x << endl;  
    cout << i * 2 << endl;  
    return 0;
```

```
}
```


Pengubahan urutan eksekusi yang merubah hasil eksekusi

```
/* contoh aksi sekuensial per  
baris */
```

```
int main()  
{  
    /* Kamus */  
    int i;  
    float x;  
  
    /* Algoritma */  
  
    cin >> i;  
    x = 100.75;  
  
    cout << x << endl;  
    cout << i * 2 << endl;  
    return 0;  
}
```

```
/* contoh aksi sekuensial per  
baris */
```

```
int main()  
{  
    /* Kamus */  
    float x;  
    int i;  
  
    /* Algoritma */  
  
    x = 100.75;  
    cin >> i;  
  
    cout << i * 2 << endl;  
    cout << x << endl;  
    return 0;  
}
```

Contoh



Buatlah program untuk menghitung jumlah dari dua buah pecahan

- Spesifikasi program
 - Program menerima masukan pecahan pertama berupa pembilang dan penyebut
 - Kemudian program menerima pecahan kedua
 - Lalu program akan melakukan penjumlahan
 - Kemudian menampilkan hasilnya berupa pembilang dan penyebut hasil penjumlahan

Contoh- Langkah Solusi 1

- Program menerima masukan pecahan pertama berupa pembilang dan penyebut
cin >> pembilang1
cin >> penyebut1
- Kemudian program menerima pecahan kedua
cin >> pembilang2
cin >> penyebut2

Contoh- Langkah Solusi 2

- Lalu program akan melakukan penjumlahan

$$\frac{A}{B} - \frac{C}{D} = \frac{AD + BC}{B * D}$$

pembilang3 = pembilang1 * penyebut2 + pembilang2 * penyebut1;

penyebut3 = penyebut1 * penyebut2;

- Kemudian menampilkan hasilnya berupa pembilang dan penyebut hasil penjumlahan

```
cout << pembilang3;
```

```
cout << penyebut3;
```

```

// Program JumlahPecahan // Menghitung pembilang dan penyebut pecahan dari penjumlahan
// dua buah pecahan yang diketahui pembilang dan penyebutnya
#include <iostream>
using namespace std;
int main() {
    // KAMUS
    int pembilang1, pembilang2, pembilang3;
    int penyebut1, penyebut2, penyebut3;
    // ALGORITMA
    cin >> pembilang1;
    cin >> penyebut1;
    cin >> pembilang2;
    cin >> penyebut2;

    pembilang3 = pembilang1 * penyebut2 + pembilang2 * penyebut1;
    penyebut3 = penyebut2 * penyebut1;

    cout << pembilang3 << endl;
    cout << penyebut3 << endl;
    return 0;
}

```

Latihan



- Buatlah program untuk Jarak dari kecepatan dan waktu suatu kendaraan
- Spesifikasi program
 - Program menerima masukan kecepatan
 - Kemudian program menerima masukan waktu
 - Lalu program akan melakukan perhitungan
 - Kemudian menampilkan hasilnya berupa perhitungan jarak

**SELAMAT
BELAJAR**