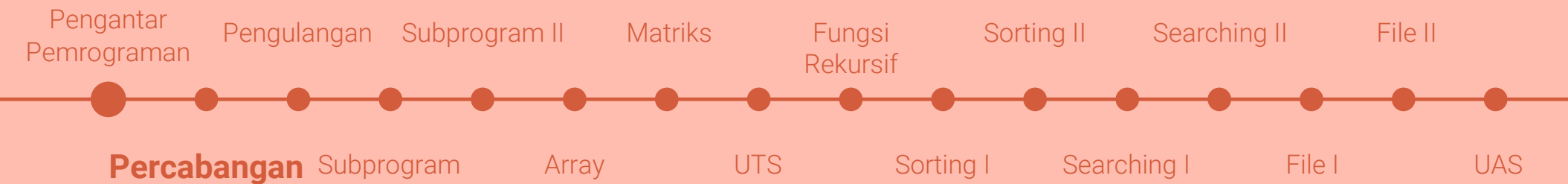


DASAR PEMROGRAMAN

Pertemuan II





Tujuan

- Mahasiswa mampu merancang algoritma penyelesaian masalah percabangan.
- Mahasiswa mampu menjelaskan tabel kebenaran dari suatu operasi Boolean.
- Mahasiswa mampu menjelaskan jenis-jenis percabangan dalam suatu bahasa pemrograman.
- Mahasiswa dapat menggunakan notasi percabangan yang sesuai dengan benar.





Materi

STRUKTUR PERCABANGAN

PILIHAN PADA PERCABANGAN

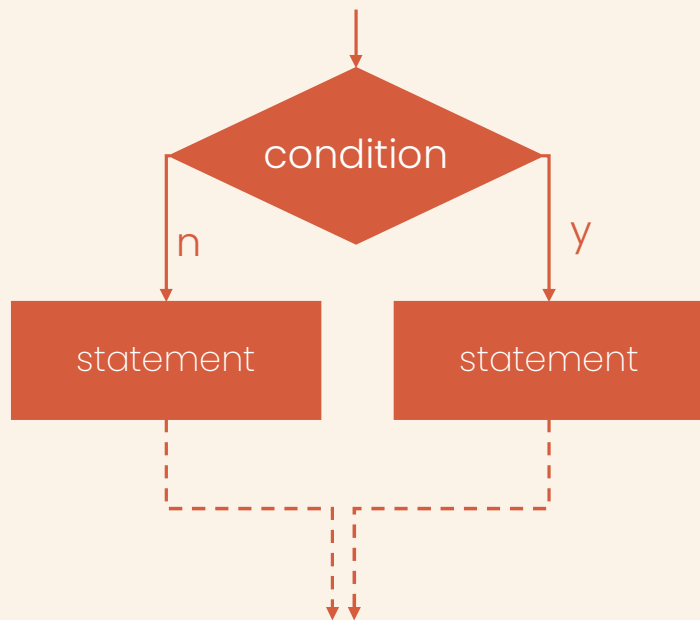
CHAINED CONDITIONAL

NESTED CONDITIONAL

KONDISI KOMPLEKS

STRUKTUR PERCABANGAN

PERCABANGAN





- Struktur algoritma dan pemrograman dimana hasil pemeriksaan suatu syarat menentukan pilihan aksi yang dilakukan berikutnya.
- Terdiri dari **kondisi/syarat** dan **aksi/perintah**

SINTAKS

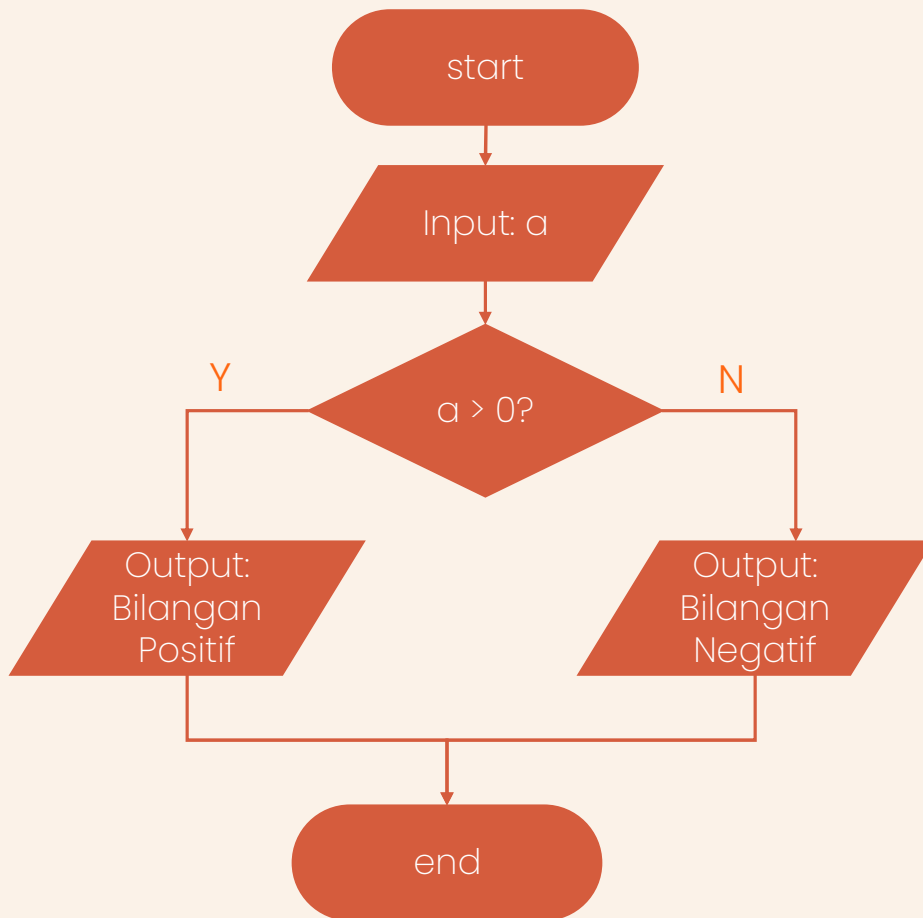
```
if (kondisi)
{
    cabang_1;
}
else
{
    cabang_2;
}
```

- Kondisi dalam percabangan harus menghasilkan nilai **Benar** atau **Salah**.
- Operasi yang bisa dilakukan pada kondisi adalah operasi relational dan operasi logika.
- Hasil pemeriksaan suatu kondisi menentukan aksi yang dilakukan

SINTAKS

```
if (kondisi)
{
 cabang_1;
}
else
{
 cabang_2;
}
```

- Aksi-aksi yang dilakukan jika hasil pemeriksaan kondisi adalah **Benar** diletakkan paling atas
- Blok else berisi aksi-aksi jika hasil pemeriksaan kondisi adalah **Salah**
- Aksi dituliskan diantara { dan } dengan posisi yang menjorok.



```
#include <iostream>

using namespace std;

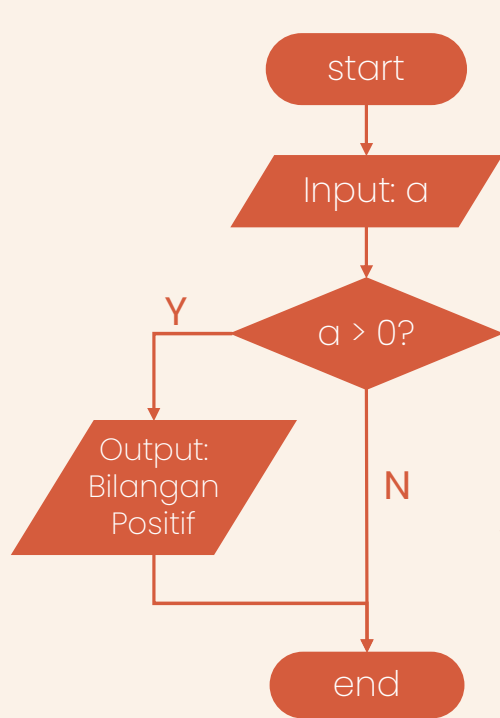
int main()
{
    int a;

    cin >> a;

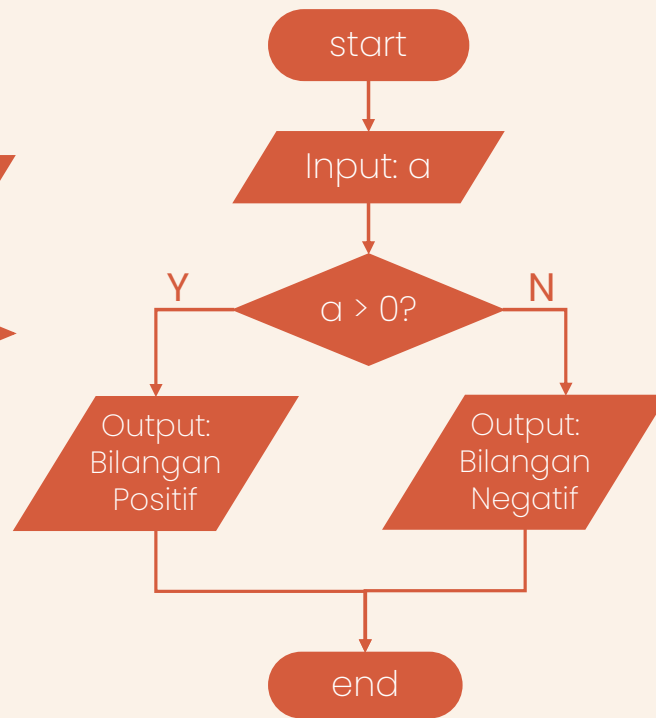
    if (a > 0)
    {
        cout << "bilangan positif";
    }
    else {
        cout << "bilangan negatif";
    }

    return 0;
}
```

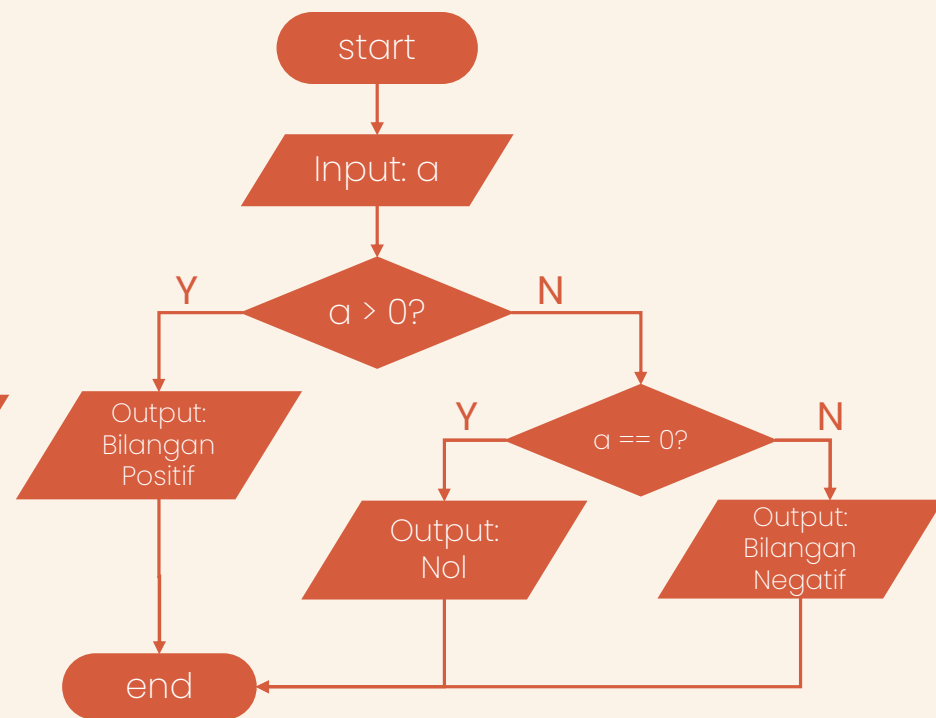
PILIHAN PERCABANGAN



1 PILIHAN



2 PILIHAN



BANYAK PILIHAN

```
#include <iostream>

using namespace std;

int main()
{
    int a;

    cin >> a;

    if (a > 0)
    {
        cout << "bilangan positif";
    }

    return 0;
}
```

1 PILIHAN

```
#include <iostream>

using namespace std;

int main()
{
    int a;

    cin >> a;

    if (a > 0)
    {
        cout << "bilangan positif";
    }
    else {
        cout << "bilangan negatif";
    }

    return 0;
}
```

2 PILIHAN

```
#include <iostream>

using namespace std;

int main()
{
    int a;

    cin >> a;

    if (a > 0)
    {
        cout << "bilangan positif";
    }
    else if (a == 0)
    {
        cout << "nol";
    }
    else {
        cout << "bilangan negatif";
    }

    return 0;
}
```

BANYAK PILIHAN



PRINSIP PERCABANGAN

- Hanya satu cabang yang akan dijalankan dalam runtunan if/else if/else.
- Percabangan yang dijalankan adalah percabangan pertama dalam runtunan yang kondisinya benar
- Percabangan pada runtunan berikutnya/dibawahnya diabaikan terlepas kondisinya benar atau salah

INDENTATION

- Kondisi/syarat dan aksi dalam percabangan harus dituliskan dalam struktur yang menjorok (*indent*).
- Bagian-bagian yang menjorok ini menandakan kapan harus keluar dari sub bagian kode tertentu.
- Bagian menjorok membedakan aksi yang dilakukan jika kondisi benar dengan aksi-aksi lainnya.

LATIHAN SOAL

LATIHAN 1

Buatlah sebuah program yang menerima suhu air (dalam derajat celcius) dan menuliskan wujud air ke layar sebagai berikut:

Jika suhu air ≤ 0 derajat, maka tuliskan "beku"



LATIHAN 2

Buatlah sebuah program yang menerima suhu air (dalam derajat celcius) dan menuliskan wujud air ke layar sebagai berikut:

- Jika suhu air ≤ 0 derajat, maka tuliskan "beku"
- Jika suhu air > 0 maka tuliskan "cair"



LATIHAN 3

Buatlah sebuah program yang menerima suhu air (dalam derajat celcius) dan menuliskan wujud air ke layar sebagai berikut:

- Jika suhu air ≤ 0 derajat, maka tuliskan "beku"
- Jika suhu air > 0 maka tuliskan "cair"
- Jika suhu air ≥ 100 , maka tuliskan "uap"





LATIHAN 4

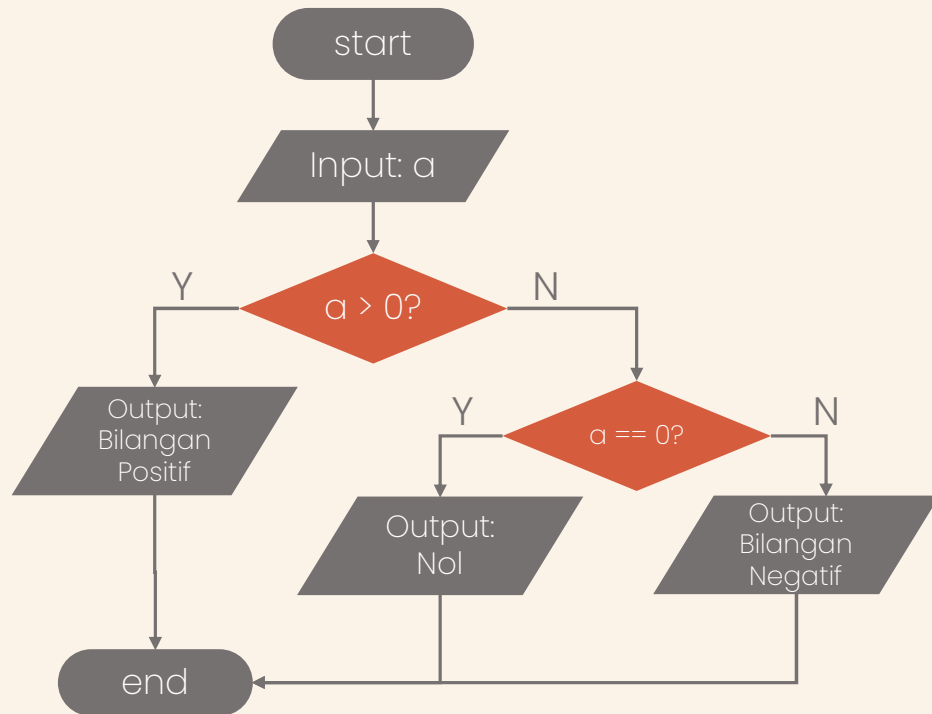
Buatlah program yang menerima masukan 2 buah integer dan menghasilkan manakah dari kedua integer tersebut yang paling besar.

Misal integernya adalah A dan B:

$A \geq B$ maka tampilkan "angka terbesar adalah A"

$A < B$ maka tampilkan "angka terbesar adalah B"

CHAINED CONDITIONAL



CHAINED CONDITIONAL

- Percabangan untuk banyak pilihan disebut juga **chained conditional** atau **chained if**.
- Disebut berantai karena masih ada syarat/kondisi lanjutan jika kondisi sebelumnya **salah**.

```

#include <iostream>

using namespace std;

int main()
{
    int a;

    cin >> a;

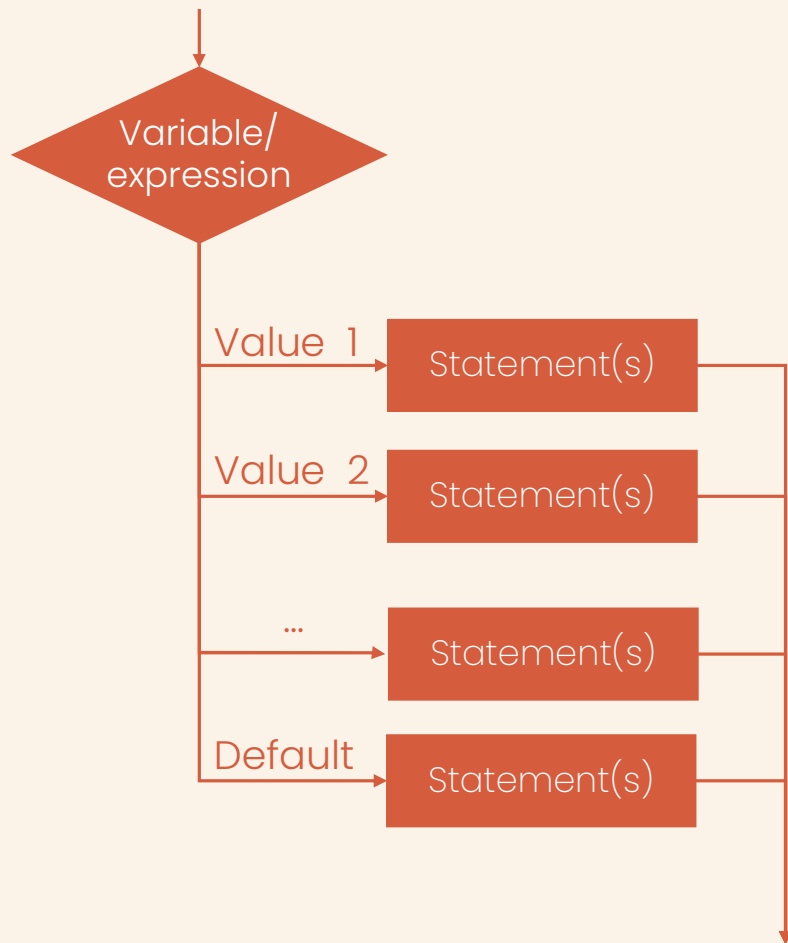
    if (a > 0)
    {
        cout << "bilangan positif";
    }
    else if (a == 0)
    {
        cout << "nol";
    }
    else {
        cout << "bilangan negatif";
    }

    return 0;
}

```

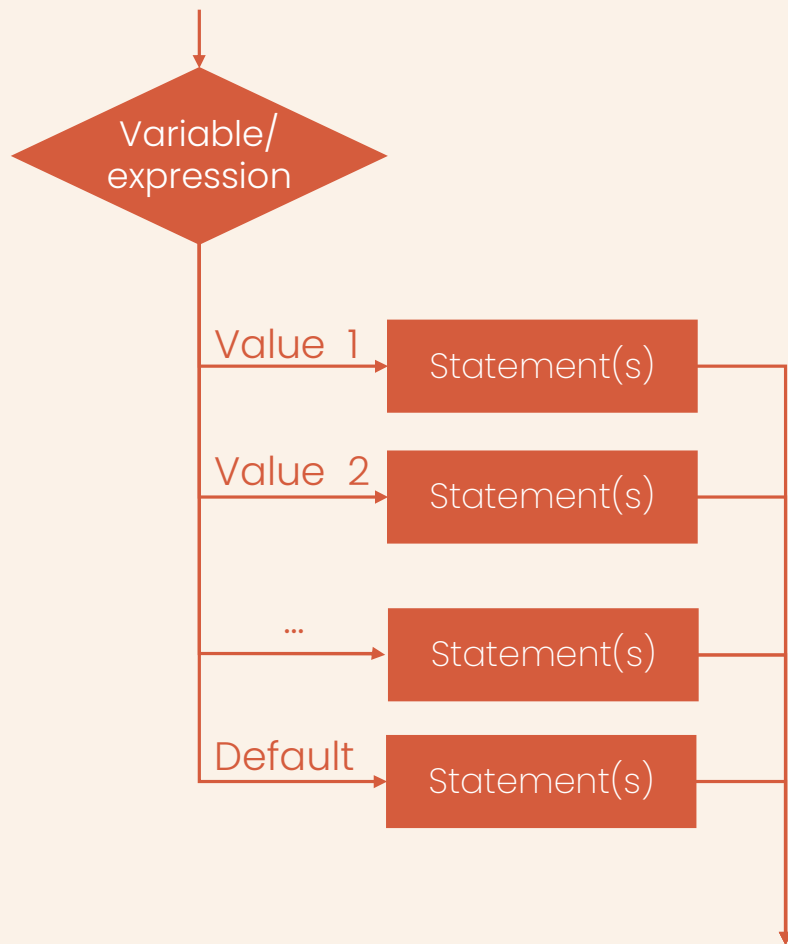
CHAINED CONDITIONAL

- Tidak ada batasan jumlah kondisi pada percabangan `else if`.
- Hanya boleh ada **satu** `else`, dan merupakan percabangan terakhir.
- Jika semua kondisi tidak terpenuhi, aksi pada percabangan `else` ini yang dijalankan.



SWITCH STATEMENT

- Digunakan untuk percabangan dengan banyak pilihan.
- Switch statement memeriksa kesamaan nilai suatu variabel/ekspresi terhadap sekumpulan nilai.



KOMPONEN SWITCH STATEMENT

- Setiap nilai pada kumpulan nilai disebut dengan **case**.
- Variabel/ekspresi pada bagian **switch** diperiksa kesamaannya dengan setiap nilai tersebut.
- **Default** adalah percabangan terakhir yang aksi pada percabangan ini akan dijalankan jika nilai pada variable/ekspresi tidak sesuai dengan case manapun
- **Break** untuk keluar dari switch statement.



Tipe data nilai pada case hanya bisa integer dan character



SINTAKS

```
switch (variable/expression)
{
    case value_1: //actions
        break;
    case value_2: //actions
        break;
    case ... : //actions
        break;
    default : //actions
        break;
}
```

CONTOH

```
#include <iostream>

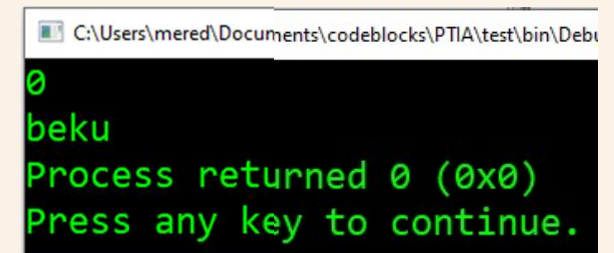
using namespace std;

int main()
{
    int suhu;

    cin >> suhu;

    switch (suhu)
    {
        case 1 : cout << "cair";
                break;
        case 0 : cout << "beku";
                break;
        case 100 : cout << "uap";
                break;
        default : cout << "tidak diketahui";
                break;
    }

    return 0;
}
```



C:\Users\mered\Documents\codeblocks\PTIA\test\bin\Debu
0
beku
Process returned 0 (0x0)
Press any key to continue.

```

#include <iostream>

using namespace std;

int main()
{
    int suhu;

    cin >> suhu;

    switch (suhu)
    {
        case 1 : cout << "cair";

        case 0 : cout << "beku";

        case 100 : cout << "uap";

        default : cout << "tidak diketahui";

    }

    return 0;
}

```



- Jika tidak ada break diakhir suatu case, aksi pada case-case berikutnya akan dijalankan hingga selesai.
- Perintah break menyuruh program untuk keluar dari bagian switch

```

C:\Users\mered\Documents\codeblocks\PTIA\test\bin\Debu
0
bekuuaptidak diketahui
Process returned 0 (0x0)
Press any key to continue.

```

NESTED CONDITIONAL

```
if (expression 1)
{
    //Executes when the expression 1 is true
    if (expression 2)
    {
        //Executes when the expression 2 is true
    }
}
else
{
    //Executes when the expression 1 is false
    if (expression 3)
    {
        //Executes when the expression 3 is true
    }
}
```

NESTED CONDITIONAL

Percabangan berlapis terjadi saat ada percabangan di dalam percabangan lain.

CONTOH

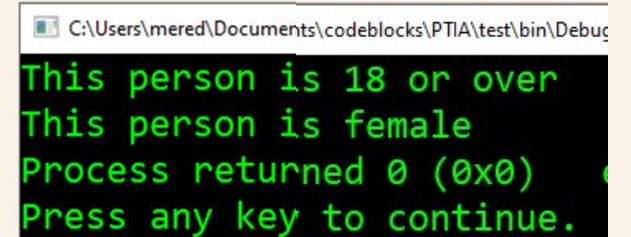
```
#include <iostream>

using namespace std;

int main()
{
    int age = 18;
    string gender = "F";

    if (age==18){
        cout << "This person is 18 or over" << endl;
        if (gender == "F") {
            cout << "This person is female";
        }
        else {
            cout << "This person is male";
        }
    }
    else {
        cout << "This person is less than 16 years old";
    }

    return 0;
}
```



C:\Users\mered\Documents\codeblocks\PTIA\test\bin\Debug
This person is 18 or over
This person is female
Process returned 0 (0x0)
Press any key to continue.

KONDISI KOMPLEKS

OPERATOR LOGIKA

Operator logika **AND** dan **OR** memungkinkan menggabungkan lebih dari satu kondisi pada syarat percabangan.

CONTOH

```
#include <iostream>

using namespace std;

int main()
{
    int a = 2;
    int b = 3;
    2 1 5 4 3
    if (!(a==3) && !(b==3)) {
        cout << "neither numbers is three";
    }

    return 0;
}
```

1

$a == 3 \rightarrow \text{False}$

2

$!(\text{False}) \rightarrow \text{True}$

3

$b == 3 \rightarrow \text{True}$

4

$!(\text{True}) \rightarrow \text{False}$

5

$\text{True} \ \&\& \ \text{False} \rightarrow \text{False}$

CONTOH

```
#include <iostream>

using namespace std;

int main()
{
    int a = 2;
    int b = 3;

    if (a==3 && b==3) {
        cout << "both numbers are three";
    }
    else if (a!=3 && b!=3){
        cout << "neither numbers is three";
    }
    else if (a==3 || b==3){
        cout << "one of the numbers is three";
    }

    return 0;
}
```

STRUKTUR PADA KONDISI

- Sama seperti matematika, tanda kurung () bisa digunakan untuk mengatur urutan eksekusi
- Logical statement menjadi lebih kompleks, sehingga perlu di-*tracing* dengan menggunakan berbagai nilai

LATIHAN SOAL

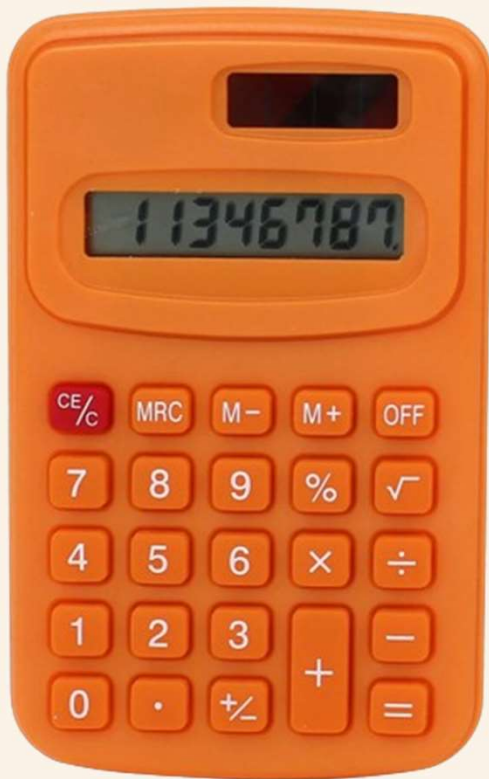


LATIHAN 1

Buatlah program yang digunakan untuk menuliskan ke layar nama-nama bulan dari nomor bulan, yaitu 1 s.d. 12 yang dibaca dari keyboard.

Program memvalidasi terlebih dahulu apakah masukan dari keyboard sudah benar atau belum, yaitu apakah nomor bulan sudah tepat 1 s.d. 12. Jika tidak tepat, maka dikeluarkan pesan kesalahan yaitu "Masukan nomor bulan tidak tepat".

LATIHAN 2



Sebuah kalkulator super sederhana, digunakan untuk menerima masukan 2 buah bilangan bulat, misalnya A dan B, dan menuliskan beberapa hasil operasi aritmatika terhadap kedua bilangan tersebut, tergantung pilihan operasi melalui character sebagai berikut:

Pilihan = a : menuliskan hasil $A + B$

Pilihan = b : menuliskan hasil $A - B$

Pilihan = c : menuliskan hasil $A * B$

Pilihan = d : menuliskan hasil $A/B \rightarrow$ hasilnya real/float

Pilihan = e : menuliskan hasil $A \text{ div } B \rightarrow$ hasilnya integer

Pilihan = f : menuliskan hasil $A \text{ mod } B$

Jika dimasukkan pilihan menu character lain, maka dituliskan "Bukan pilihan menu yang benar"

Masukan		Pilihan Operasi	Keluaran
A	B		
3	2	A	5
3	2	B	1
3	2	C	6
3	2	D	1.5
3	2	E	1
3	2	F	1
3	2	G	Bukan pilihan menu yang benar

