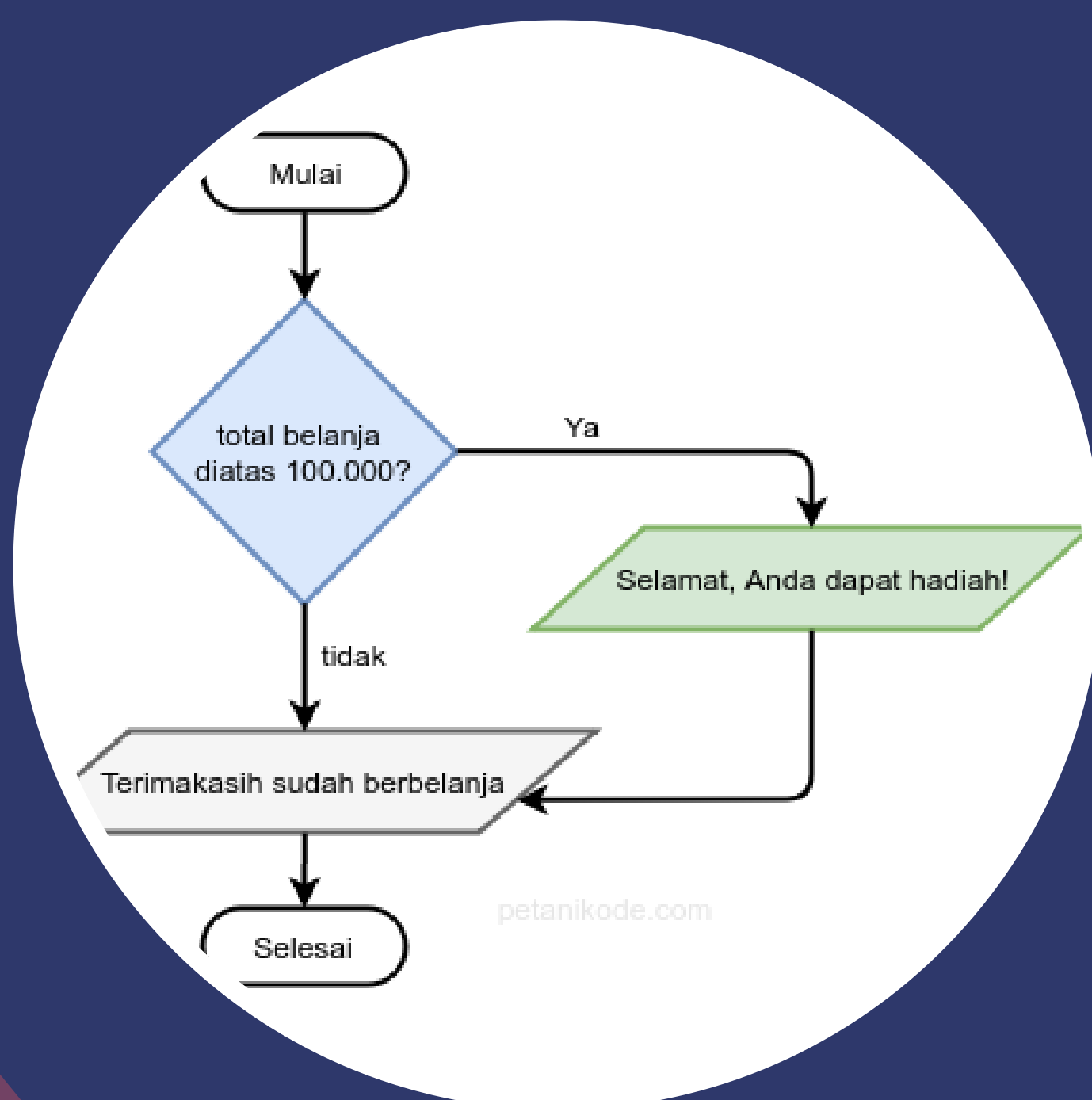


MODUL PERCABANGAN

Modul Percabangan Bagian I



Disusun Oleh :
**Johannes Alexander
Putra**

Capaian Pembelajaran

Pada akhir fase E, peserta didik mampu menerapkan praktik baik konsep pemrograman prosedural dalam salah satu bahasa pemrograman prosedural dan mampu mengembangkan program yang terstruktur dalam notasi algoritma atau notasi lain, berdasarkan strategi algoritmik yang tepat.

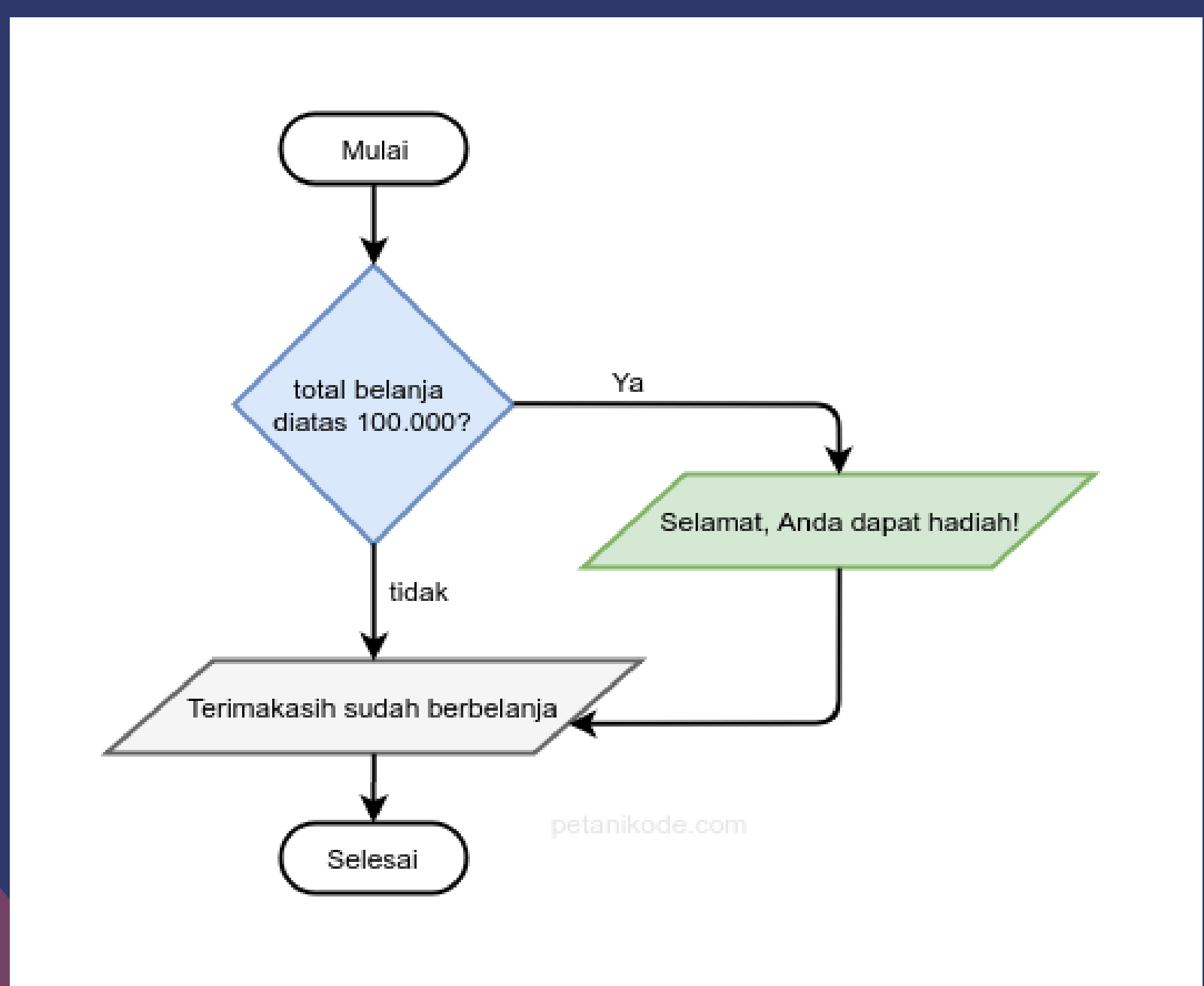
Tujuan Pembelajaran

- Memahami permasalahan yang berkaitan dengan jenis-jenis percabangan
- Menentukan pemecahan masalah dengan jenis-jenis percabangan
- Melakukan implementasi jenis-jenis percabangan untuk suatu permasalahan
- Melakukan evaluasi terhadap penggunaan jenis-jenis percabangan
- Memahami permasalahan yang berkaitan dengan percabangan tunggal
- Menentukan pemecahan masalah dengan percabangan tunggal
- Melakukan implementasi flowchart untuk suatu percabangan tunggal
- Melakukan evaluasi terhadap penggunaan percabangan tunggal

P e r c a b a n g a n

Suatu program yang hanya berisi runtutan instruksi biasanya terdapat dalam persoalan sederhana. Persoalan yang lebih kompleks melibatkan analisis berbagai kasus kemungkinan yang terdapat di dalamnya. Untuk setiap kasus terdapat persyaratan yang harus dipenuhi dan aksi apa yang harus dilakukan jika persyaratan dipenuhi. Dengan adanya analisis kasus maka instruksi tidak dikerjakan secara sekuensial lagi akan tetapi melakukan aksi jika persyaratan dipenuhi (Munir & Lidya, 2016).

Percabangan If merupakan sebuah blok program yang menyatakan bahwa sebuah aksi akan dijalankan jika kondisi percabangan dipenuhi jika tidak dipenuhi maka aksinya tidak akan dijalankan. Percabangan if biasa digunakan untuk mengerjakan aksi yang memiliki syarat tertentu (Sukamto, 2018)



G a m b a r a n

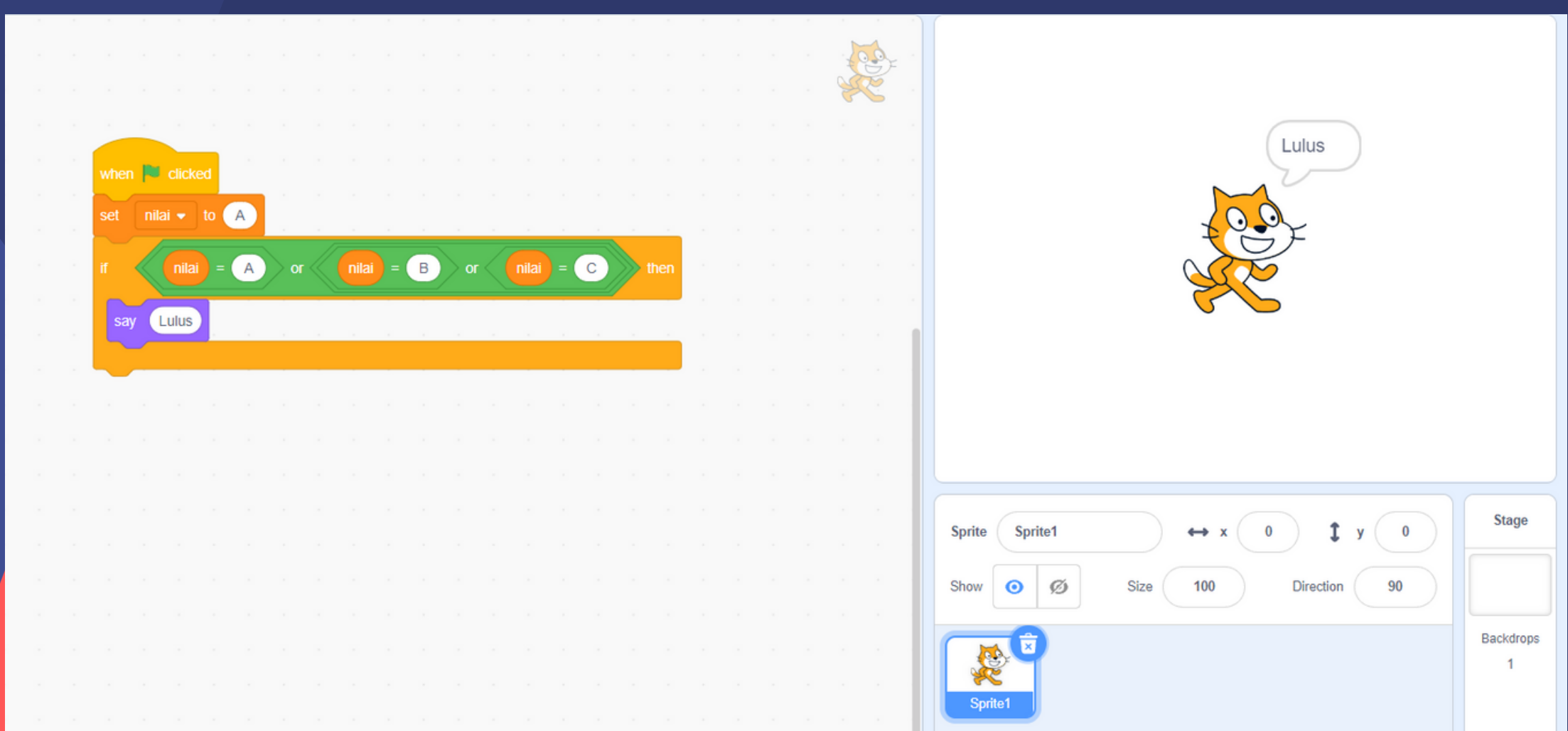
P e r c a b a n g a n

Misalkan ada sebuah kotak yang berisi nilai. Jika kotak tersebut menampung nilai A/B/C maka akan mencetak lulus. Dalam kasus ini kotak tersebut menampung nilai A. maka dapat disimpulkan bahwa kotak tersebut menampung nilai lulus. Maka dapat dituliskan:

```
#include <iostream>

using namespace std;

int main()
{
    char nilai = 'b';
    if (nilai == 'a' || nilai == 'b' || nilai == 'c'){
        cout<<"Lulus";
    }
}
```



Kasus lain

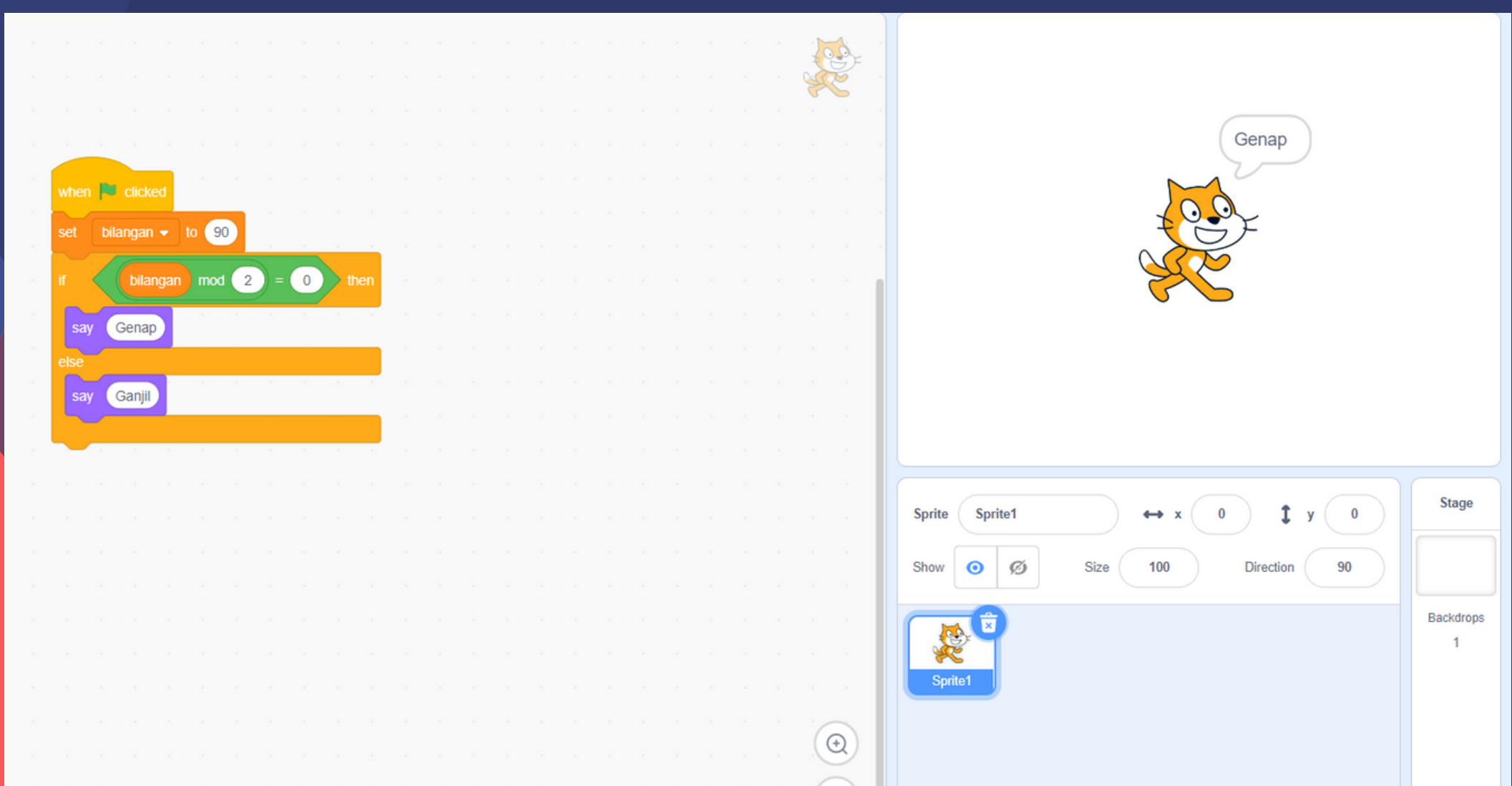
Percabangan dua kasus memiliki 2 kasus. Pertama kasus jika syarat tidak terpenuhi dan kasus kedua jika syarat terpenuhi. Misalnya pengecekan apakah suatu bilangan ganjil atau genap

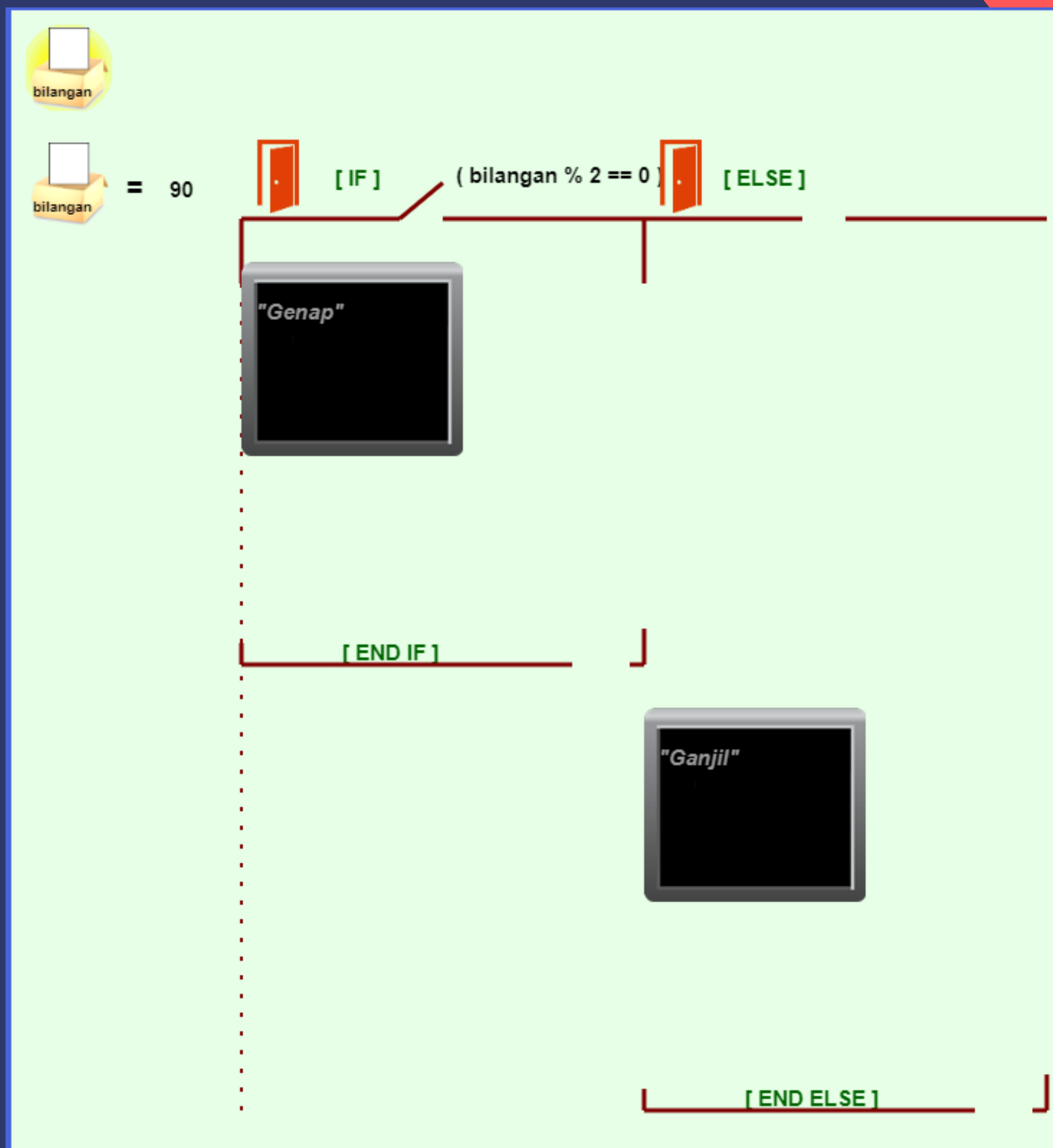
```
#include <iostream>

using namespace std;
int main(){

    int bilangan;
    bilangan = 90
    if(bilangan % 2 == 0){
        cout<<"Genap";
    }else{
        cout<<"Ganjil";
    }

}
```





Misalnya dalam kasus tersebut bilangan diisi dengan angka 90. Kita ketahui bahwa 90 merupakan bilangan Genap. dituliskan jika sisa bagi 90 jika dibagi 2 adalah 0 maka genap, dan jika tidak bilangan tersebut adalah ganjil. Tentu kita ketahui bahwa $90/2$ menghasilkan sisa bagi 0. Kita bisa menebak hasilnya adalah Genap karena memenuhi kondisi pada if

Bagaimana jika nilainya adalah 91. Kita lihat kondisinya Jika sisa bagi nilainya adalah 0 maka Genap. Sedangkan $91/2$ menghasilkan sisa bagi 1. Maka aksi yang dilaksanakan adalah aksi pada else

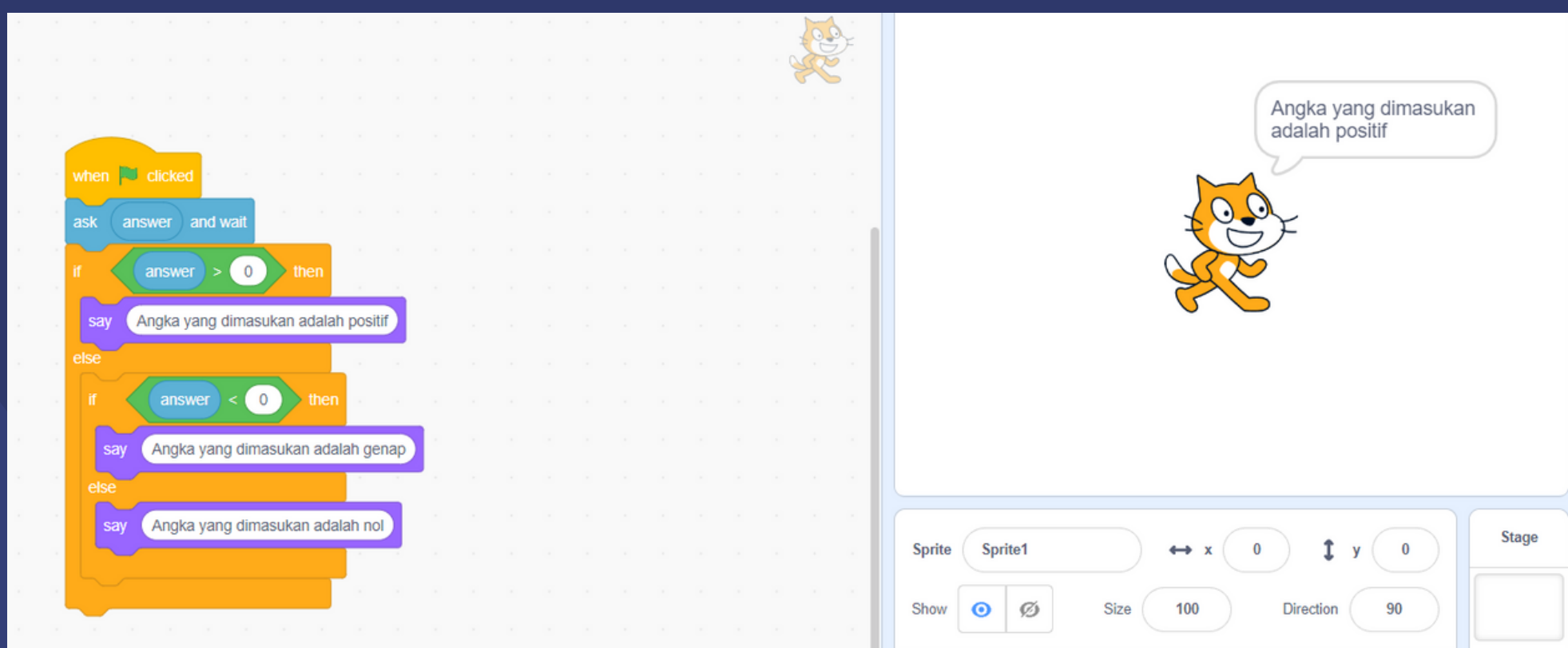
Kasus lain

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

int main() {
    int angka;
    cout << "Masukkan sebuah angka: ";
    cin >> angka;

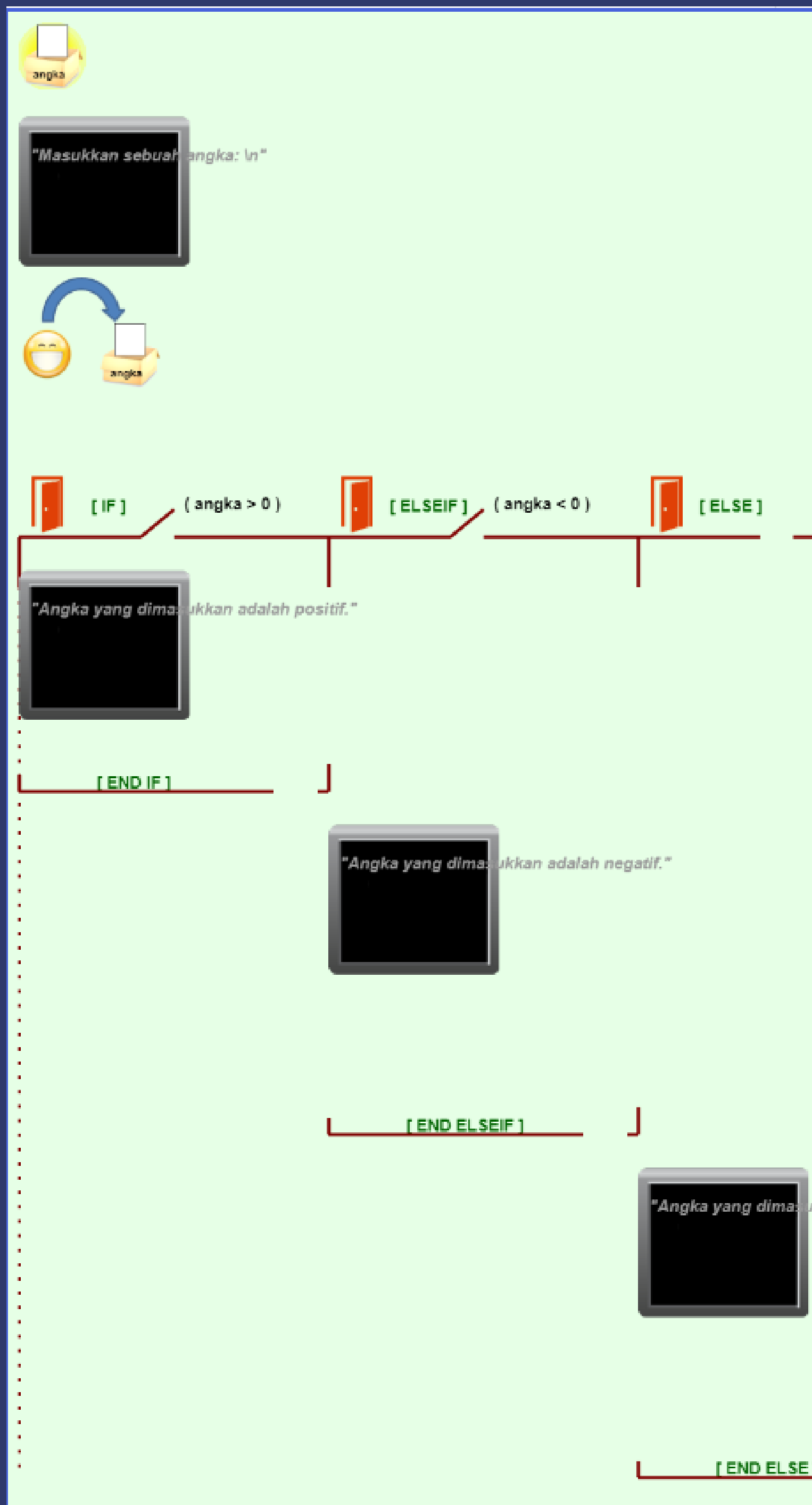
    if (angka > 0) {
        cout << "Angka yang dimasukkan adalah positif." << endl;
    } else if (angka < 0) {
        cout << "Angka yang dimasukkan adalah negatif." << endl;
    } else {
        cout << "Angka yang dimasukkan adalah nol." << endl;
    }

    return 0;
}
```



Pada contoh di atas, kita meminta pengguna untuk memasukkan sebuah angka. Kemudian, kita menggunakan percabangan tiga kasus (if, else if, dan else) untuk mengecek apakah angka tersebut positif, negatif, atau nol. Pesan yang sesuai akan ditampilkan di layar berdasarkan nilai yang dimasukkan oleh pengguna.

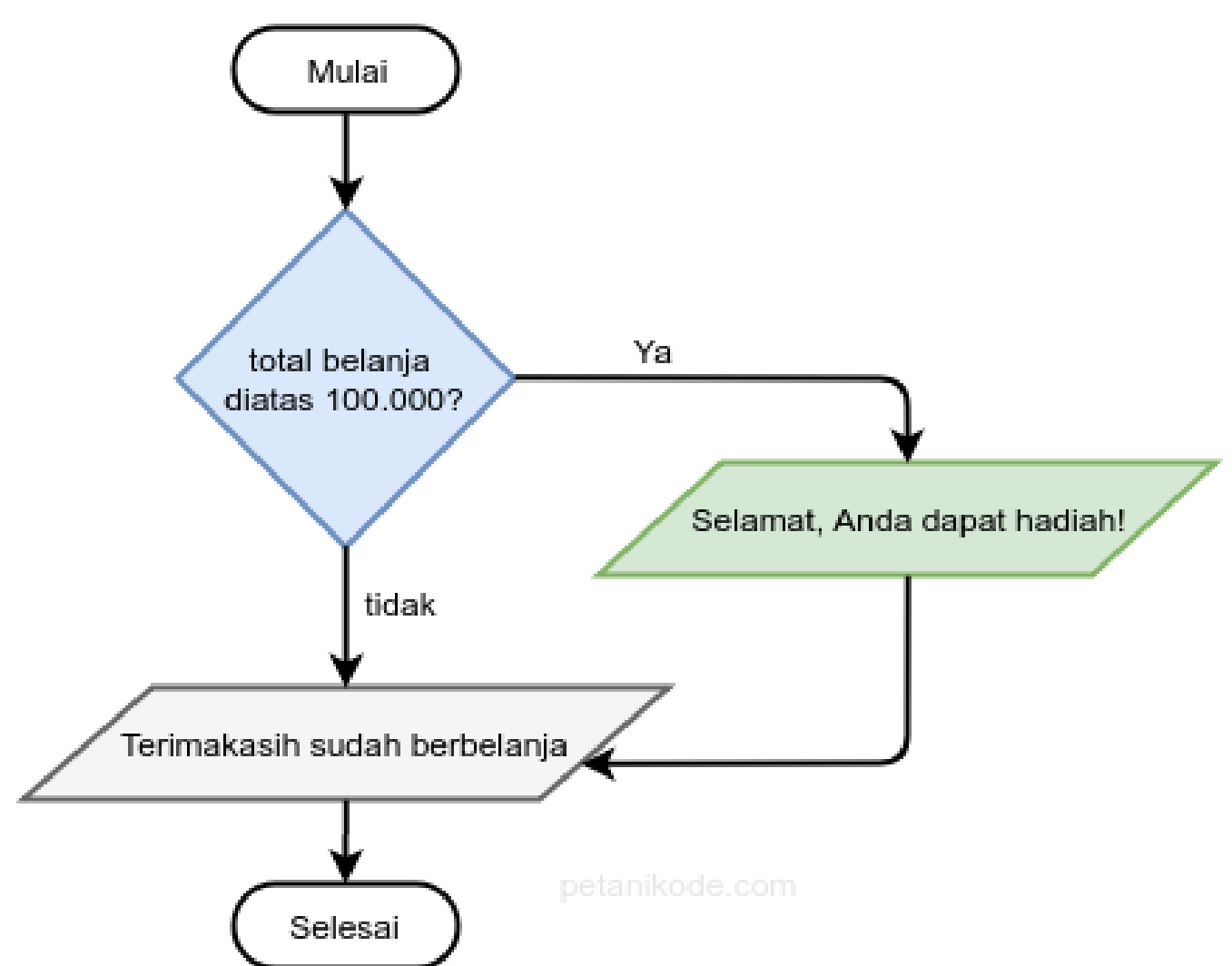
Jika angka lebih besar dari nol, maka program akan menampilkan "Angka yang dimasukkan adalah positif." Jika angka kurang dari nol, program akan menampilkan "Angka yang dimasukkan adalah negatif." Jika angka sama dengan nol, program akan menampilkan "Angka yang dimasukkan adalah nol." Misalnya dalam kasus ini kita masukan angka > 0



Pada Kasus di atas jika memasukan angka 10 maka akan mencetak Angka yang dimasukan adalah positif. Sedangkan jika memasukan -10 maka akan mencetak angka yang dimasukan adalah negatif. Sedangkan jika memasukan 0 maka akan mencetak angka yang dimasukan nol. Hal ini disebabkan 10 memenuhi kondisi if pertama karena $10 > 0$. Sedangkan -10 memenuhi kondisi else if karena $-10 < 0$. dan 0 memenuhi kondisi else karena kondisi if dan kondisi else if tidak terpenuhi maka blok else yang dijalankan

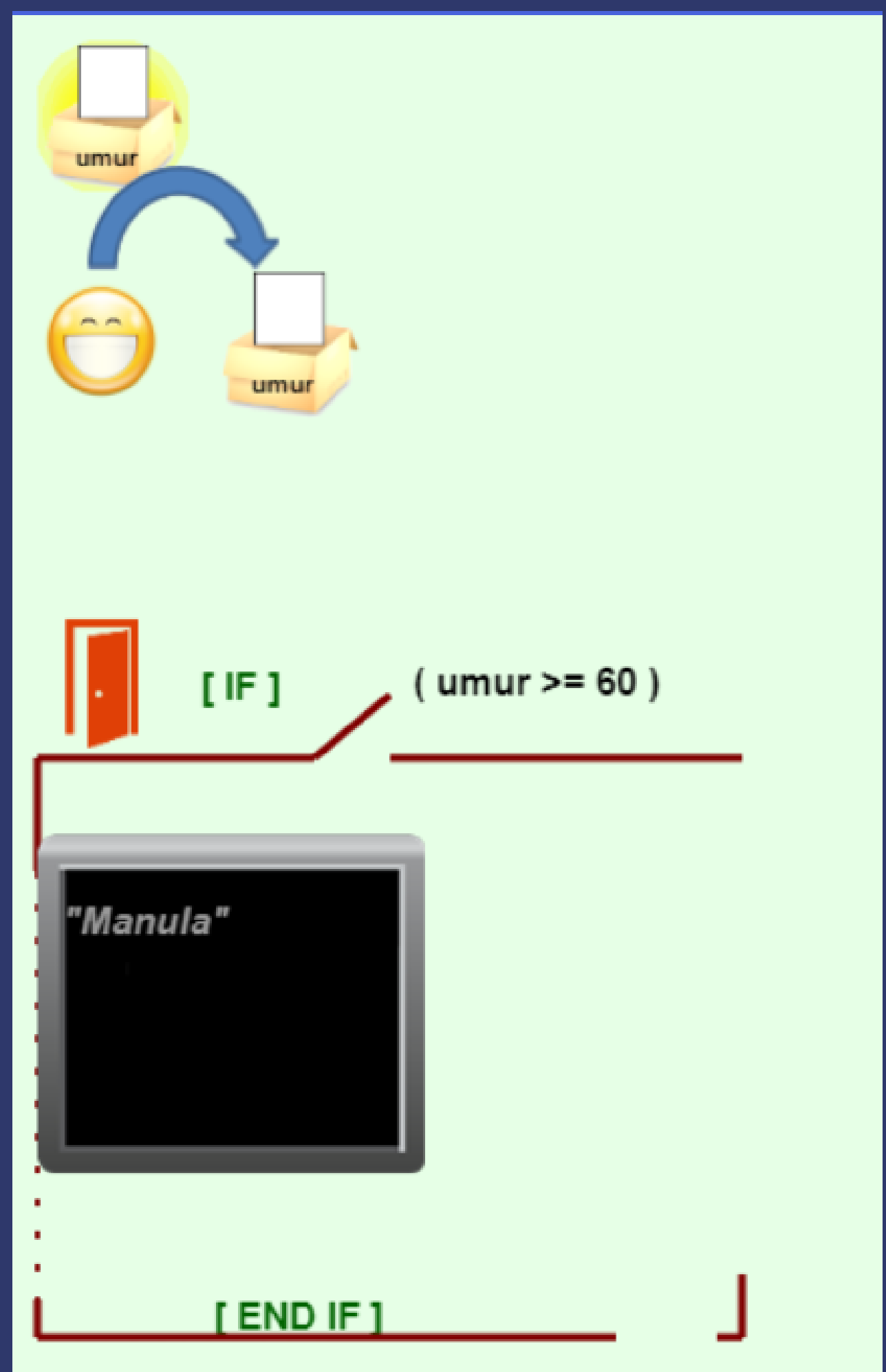
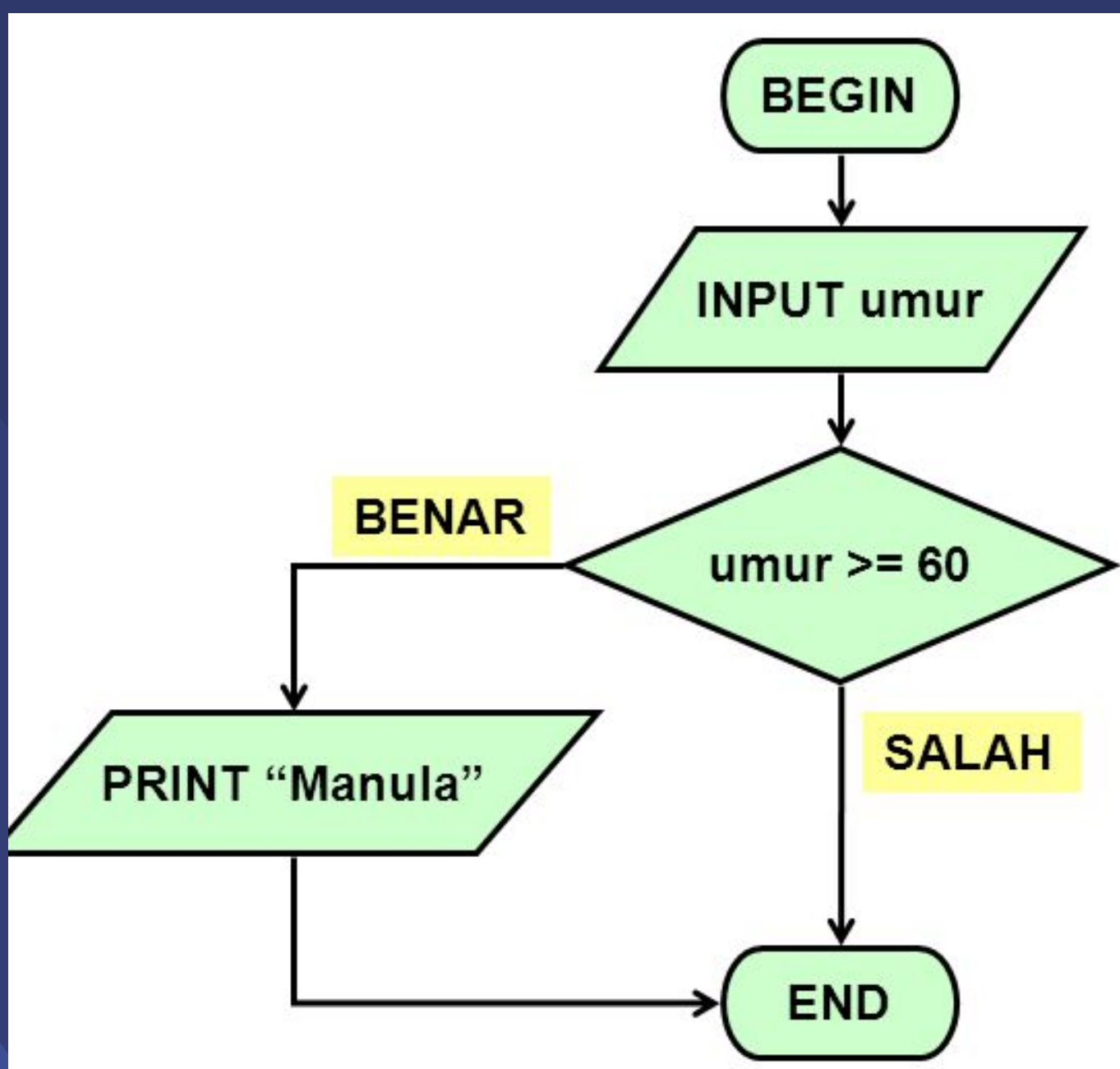
Dapat disimpulkan Jenis-Jenis Percabangan

- Percabangan Tunggal
Jika hanya ada satu kondisi
- Percabangan Dua Kasus
Jika hanya ada 2 kondisi
- Percabangan Tiga Kasus atau lebih
Jika hanya ada 3 kasus atau lebih
- Switch
Akan dibahas di pertemuan kedua



Percabangan Tunggal

Blok program if untuk syarat satu kondisi berarti **hanya ada sebuah blok aksi yang akan dikerjakan** jika syarat kondisi terpenuhi. Jika kondisi yang ada pada jalannya program memenuhi syarat pada blok if, maka proses di dalam blok if akan dijalankan namun jika tidak memenuhi syarat blok if maka jalannya program tidak akan melakukan apa-apa



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

int main() {
    int umur;
    cin>>umur;
    if(umur >= 60){
        cout<<"Manula";
    }

    return 0;
}
```

Dalam kasus ini jika umur yang dimasukan adalah 60 atau lebih besar dari 60 maka akan memprint manula sedangkan jika tidak maka tidak melakukan apa-apa

Contoh Soal dan Penyelesaian

Jika siswa lulus SBMPTN maka siswa tersebut akan diberikan mainan oleh orang tuanya. Kira-kira apa algoritma yang sesuai?

Ada 4 Tahapan Penyelesaian masalah menurut (Polya,2015) Memahami Masalah, Merencanakan Penyelesaian Masalah, Melaksanakan Rencana Penyelesaian Masalah, dan Memeriksa Kembali

Tahap 1 (Memahami Masalah):

1. Apakah informasi yang diberikan sudah cukup untuk menjawab pertanyaan?

Jawab: Sudah cukup, karena sudah menggambarkan sebuah kondisi

2. Apakah saja data yang diketahui?

Jawab: Jika siswa tersebut lulus SBMPTN maka diberikan mainan oleh orang tuanya.

Contoh Soal dan Penyelesaian

Tahap 2 (Membuat Rencana Penyelesaian Masalah):

1. Kira-kira teorema atau kondisi apakah yang mirip dengan kasus tersebut?

Jawab: kondisi yang mirip adalah percabangan tunggal, karena hanya ada **satu aksi saja yang dilakukan**

Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

1. Jika dibuat dalam algoritma kira-kira bagaimana?

Jawab:

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

int main() {
    bool lulus;
    cin>>lulus;
    if(lulus == 1){
        cout<<"Siswa mendapatkan mainan dari orang tuanya";
    }

    return 0;
}
```

Contoh Soal dan Penyelesaian

Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

2. Bagaimana Outputnya jika dimasukan lulus adalah 1?

Jawab: kita perlu ingat bahwa 1 adalah true. Maka jika lulus adalah 1 / true maka akan melakukan aksi dan jika yang dimasukan adalah 0/false maka tidak dilakukan aksi

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

int main() {
    bool lulus;
    cin>>lulus;
    if(lulus == 1){
        cout<<"Siswa mendapatkan mainan dari orang tuanya";
    }

    return 0;
}
```

Tahap 4 (Memeriksa Kembali):

1. Apakah kode tersebut sudah benar dan menghasilkan output dengan benar. Misal kodenya begini ?:

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

int main() {
    bool lulus;
    cin>>lulus;
    if(lulus == 0){
        cout<<"Siswa mendapatkan mainan dari orang tuanya";
    }

    return 0;
}
```

Jawab: Kode tersebut tidak menghasilkan error namun, pengecekan kondisinya salah seharusnya jika lulus == 1 artinya jika lulus adalah true maka melakukan aksi: Siswa mendapatkan mainan

Daftar Pustaka

Polya, G. (2015). How to Solve It: A New Aspect of Mathematical Method. New Jersey: Princeton University Press.

Sukanto, R. A. (2018). Logika Algoritma dan Pemograman Dasar. Bandung: Modula.

Munir, R. & Lidya, L. (2016). Algoritma dan Pemrograman dalam Bahasa Pascal, C, dan C++. Bandung: Informatika