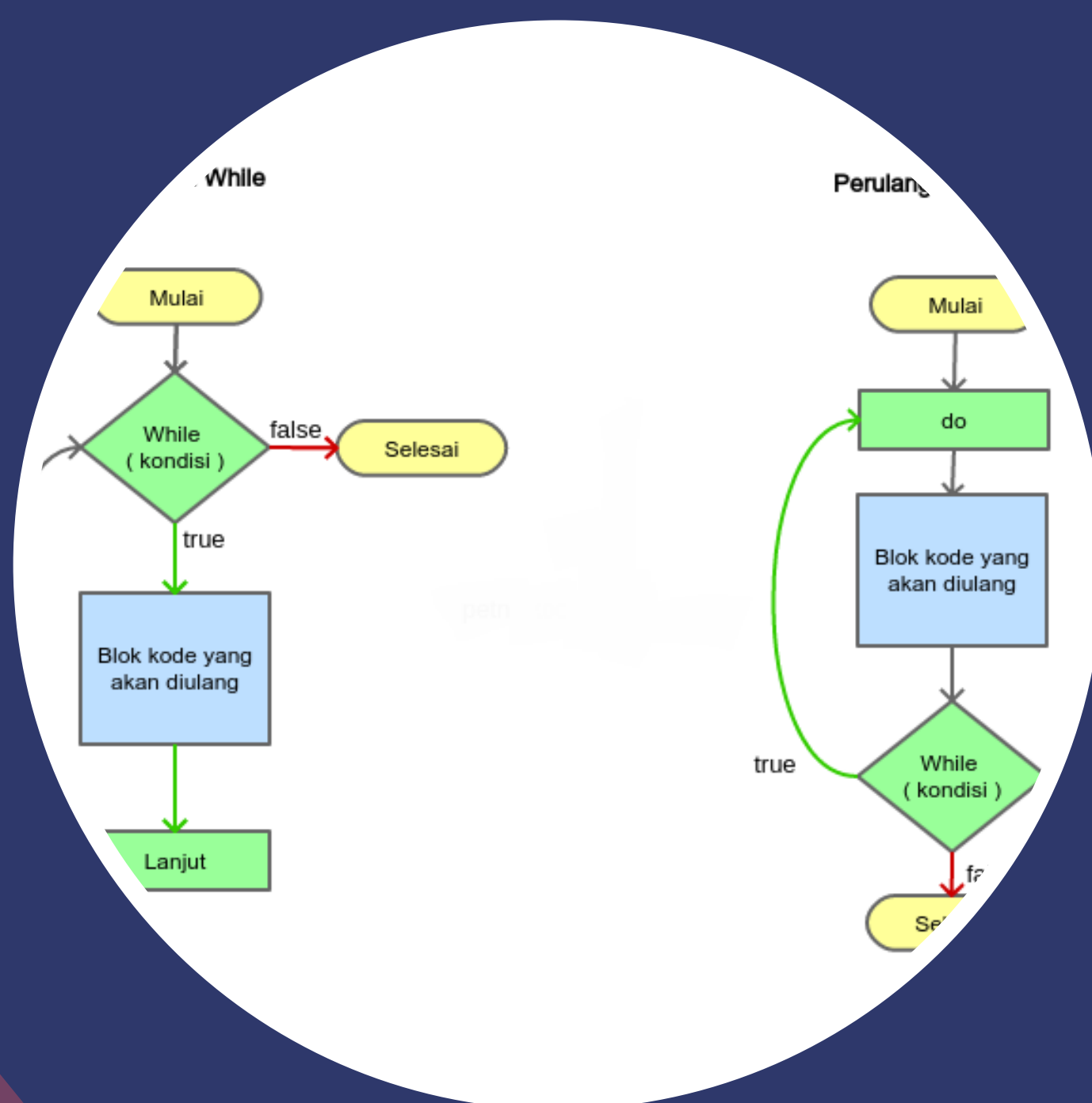


# MODUL PERULANGAN

Modul Perulangan Bagian II



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 perulangan while
- Menentukan pemecahan masalah dengan perulangan while
- Melakukan implementasi perulangan while untuk suatu permasalahan
- Melakukan evaluasi terhadap penggunaan perulangan while
- Memahami permasalahan yang berkaitan dengan perulangan while
- Menentukan pemecahan masalah dengan perulangan do while
- Melakukan implementasi perulangan do while untuk suatu permasalahan
- Melakukan evaluasi terhadap penggunaan perulangan do while

# Perulangan While

Perulangan while biasa digunakan jika jumlah perulangan tidak diketahui atau memiliki kemungkinan dapat dilakukan kurang dari batas perulangan. Perulangan while ini akan melakukan perulangan selama kondisi perulangan terpenuhi. (Sukamto, 2018)

```
#include <iostream>

using namespace std;

int main()
{
    int a = 10;
    while(a % 2 == 0){
        cout<<"Genap"<<endl;
        a--;
    }
}
```

Penjelasan:

Sebelum memasuki badan perulangan kondisi terlebih dahulu diperiksa apakah memenuhi atau tidak, Aksi dikerjakan berulang kali selama kondisinya bernilai benar. Jika kondisi bernilai salah maka kondisi tidak dikerjakan atau berhenti dikerjakan (Munir, R. & Leony,L. 2016)

Dalam kasus tersebut outputnya adalah  
Genap

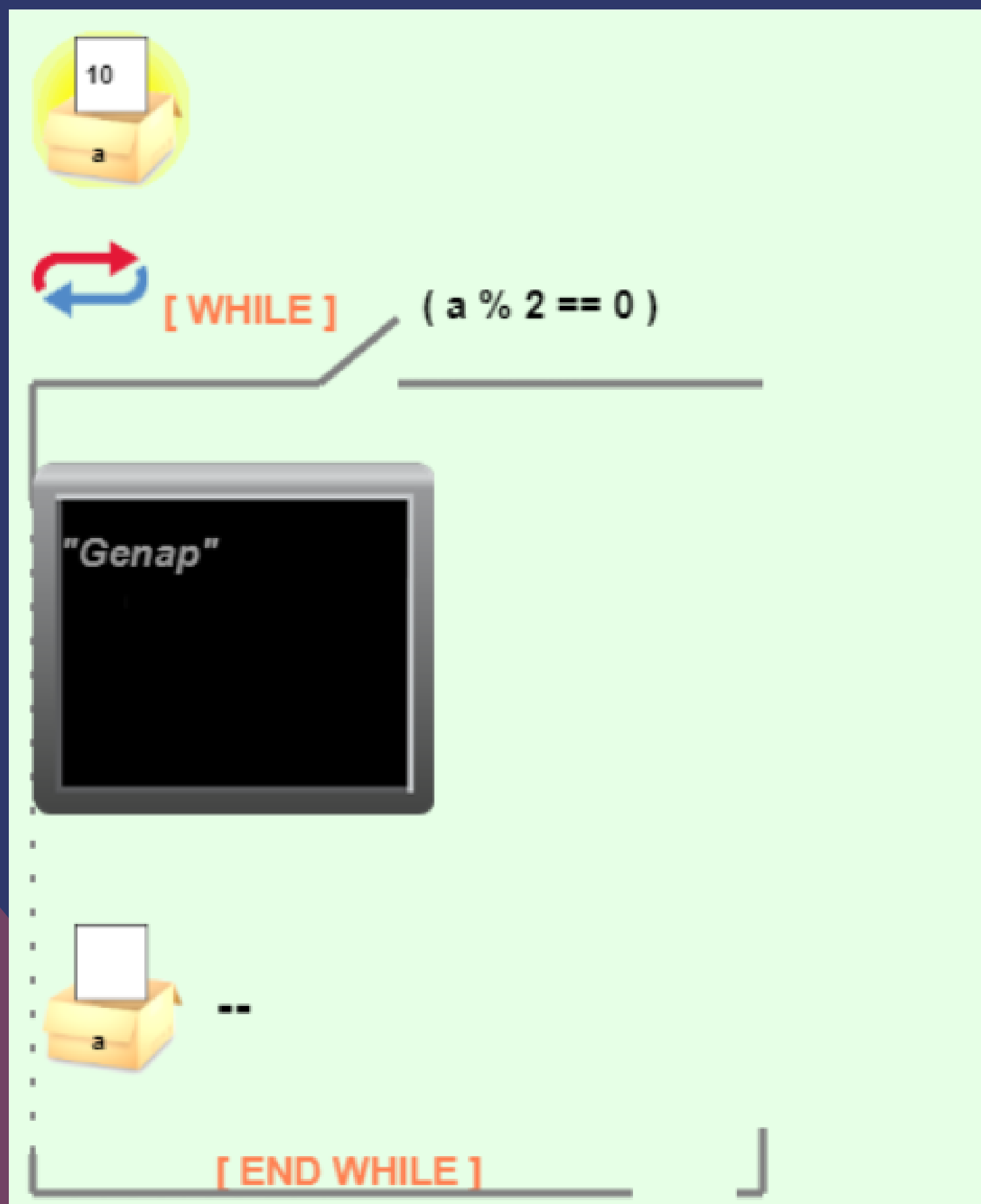
kenapa hanya satu kali diulang, karena pada perulangan kedua kondisi sudah tidak dipenuhi karena  $a = 9$  dan  $9 \% 2$  tidak sama dengan 0

```

9  #include <iostream>
10
11  using namespace std;
12
13  int main()
14  {
15      int a = 9;
16      while(a % 2 == 0){
17          cout<<"Genap"<<endl;
18          a--;
19      }
20  }

```

Jika a adalah 9 maka tidak ada aksi yang dikerjakan karena tidak memenuhi pengecekan kondisi  $a \% 2 == 0$  karena  $9 \% 2$  adalah 1



# Contoh Soal dan Penyelesaian

Miftah ingin membuat suatu aplikasi yang menerapkan prinsip perulangan yang iterasinya berdasarkan kondisi dan tidak hanya sekedar angka awal dan angka akhir saja, apakah informasi sudah cukup?

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: Belum cukup karena bisa jadi do while bisa jadi while

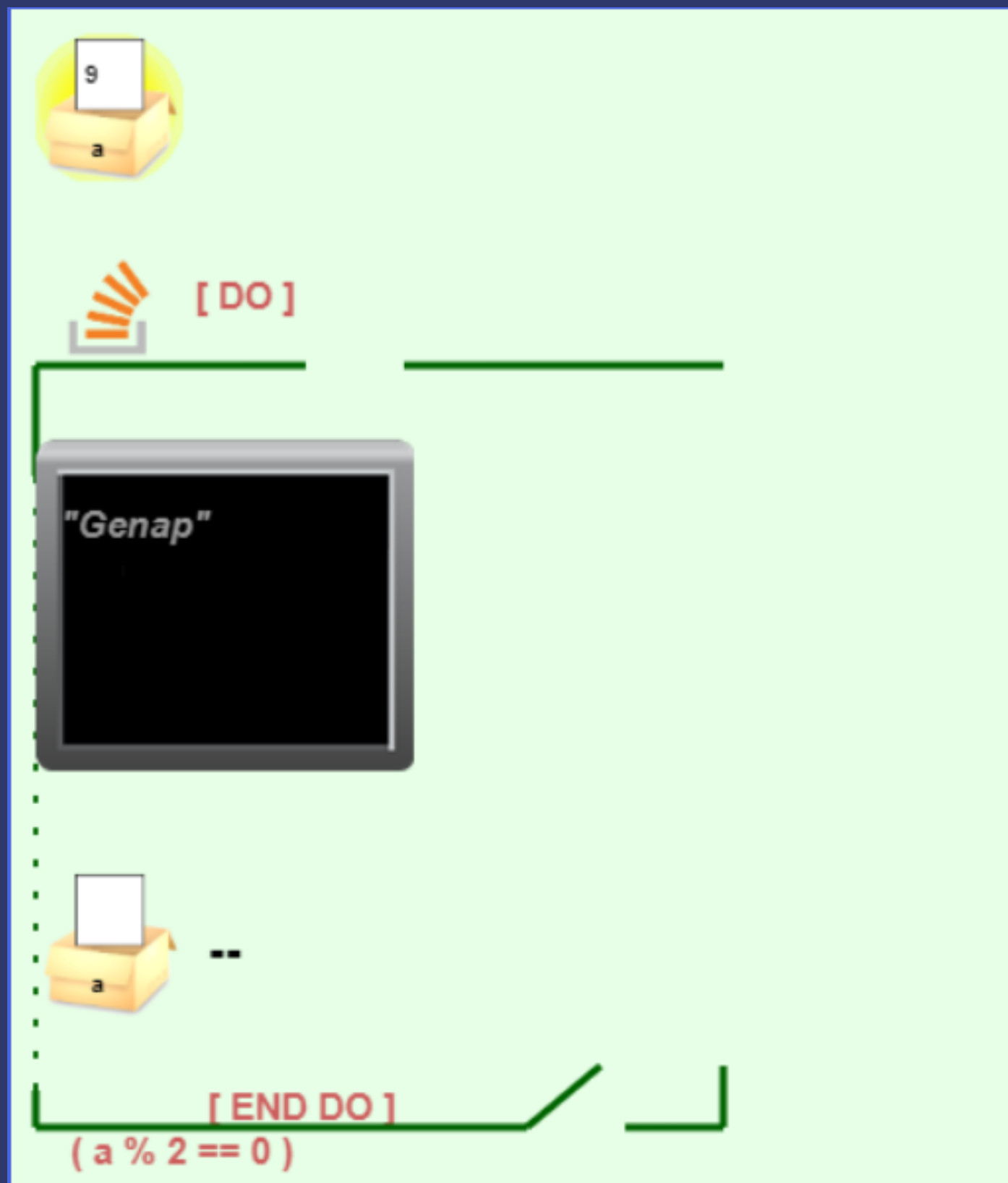
# Perulangan Do While

Perulangan Do While biasanya digunakan jika jumlah perulangannya tidak diketahui namun berbeda dengan while. karena kondisi perulangan ada di bagian bawah. Perulangan ini minimal dilakukan satu kali (Sukamto, 2018)

```
9  #include <iostream>
10
11  using namespace std;
12
13  int main()
14  {
15      int a = 9;
16      do{
17          cout<<"Genap"<<endl;
18          a--;
19      }while(a % 2 == 0);
20  }
```

## Penjelasan

Konstruksi do while/repeat mendasarkan pengulangan pada kondisi bernilai boolean. Pemeriksaan kondisi dilakukan di akhir setiap pengulangan.



Maka output yang akan dicetak adalah  
Genap  
Genap

Karena proses dikerjakan terlebih dahulu walaupun 9 bukan bilangan genap.



# Contoh Soal

Miftah ingin membuat suatu aplikasi yang menerapkan prinsip perulangan yang iterasinya berdasarkan kondisi dan tidak hanya sekadar angka awal dan angka akhir saja dan ingin mengulang minimal satu kali, yaitu ingin mencetak berdasarkan kondisi jika angka yang dimasukan adalah bilangan genap maka akan mencetak “Genap” dan iterasi selalu berkurang satu, dan batas looping adalah ketika bilangan bernilai ganjil apakah informasi sudah cukup?

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

Tahap 2 (Membuat Rencana Penyelesaian Masalah):

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

Jawab: kondisi yang mirip adalah perulangan do while karena kondisi pengecekan dilakukan di akhir.

Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

1. Buatlah penyelesaian masalah

```
9  #include <iostream>
10
11  using namespace std;
12
13  int main()
14  {
15      int a;
16      cin>> a;
17      do{
18
19          cout<<"Genap";
20          a--;
21      }while(a % 2 == 0);
22
23  }
```

Tahap 4 (Memeriksa Kembali):

1. Apakah kode tersebut sudah benar dan menghasilkan output dengan benar.

Berdasarkan kasus di atas kode sudah tepat dan penulisannya sudah tepat dapat dijalankan tanpa error.

# 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. & Leony, L. (2016). Algoritma dan Pemrograman Dalam Bahasa Pascal, C, dan C++ Edisi keenam. Bandung: Informatika