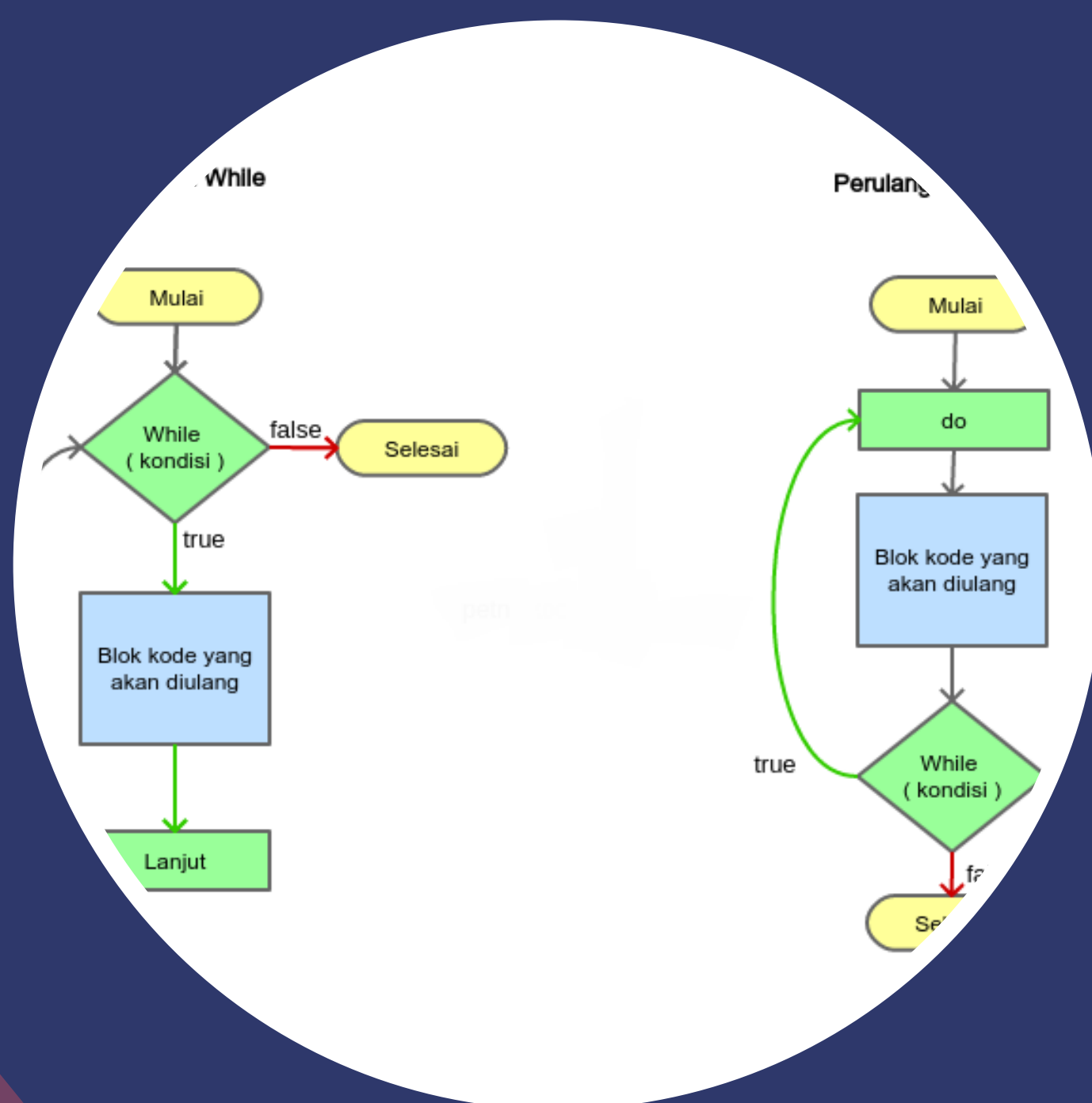


MODUL PERULANGAN

Modul Perulangan 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 perulangan
- Menentukan pemecahan masalah dengan jenis-jenis perulangan
- Melakukan implementasi jenis-jenis perulangan untuk suatu permasalahan
- Melakukan evaluasi terhadap penggunaan jenis-jenis perulangan
- Memahami permasalahan yang berkaitan dengan perulangan for
- Menentukan pemecahan masalah dengan perulangan for
- Melakukan implementasi perulangan for untuk suatu permasalahan
- Melakukan evaluasi terhadap penggunaan perulangan for

Perulangan

Perulangan (Looping) merupakan suatu bagian yang bertugas untuk melakukan kegiatan mengulang suatu proses sesuai dengan yang diinginkan. Banyak dari aplikasi perangkat lunak yang melakukan pekerjaan berulang-ulang sampai sebuah kondisi yang diinginkan, oleh karena itu pengulangan merupakan bagian penting dalam pemrograman (Sukamto, 2018)

Pengulangan memiliki bagian yang harus dipenuhi:

- Inisialisasi

Inisialisasi merupakan tahap persiapan membuat kondisi awal sebelum melakukan perulangan, misalnya dengan mengisi variable dengan nilai awal

- Proses

Tahap proses terjadi di dalam bagian perulangan di mana berisi semua proses yang perlu dilakukan secara berulang-ulang

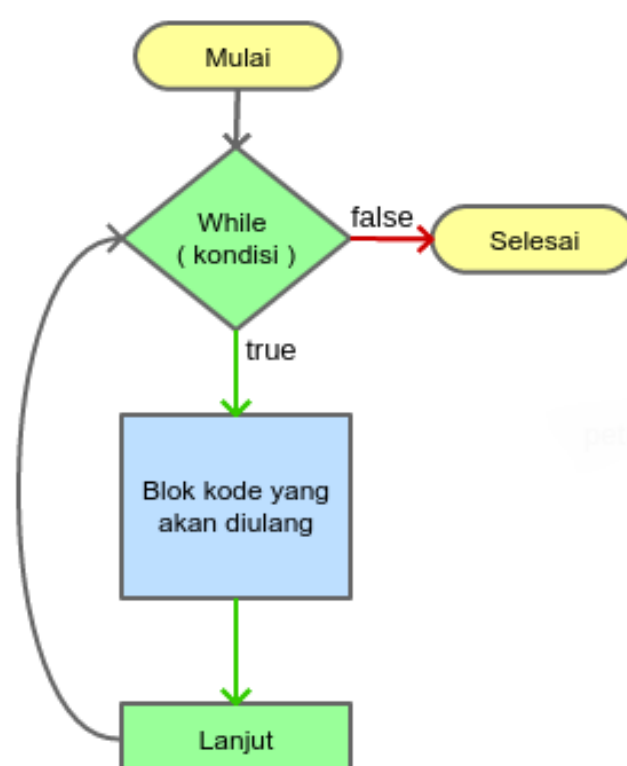
- Iterasi

Iterasi terjadi di dalam perulangan di mana merupakan kondisi pertambahan agar perulangan dapat terus berjalan

- Terminasi

Terminasi adalah kondisi berhenti dari perulangan, kondisi berhenti sangatlah penting dalam perulangan agar perulangan dapat berhenti. (Sukamto, 2018)

Perulangan While



Perulangan Do/While



Jenis Perulangan

Terdapat beberapa jenis perulangan:

- Perulangan For

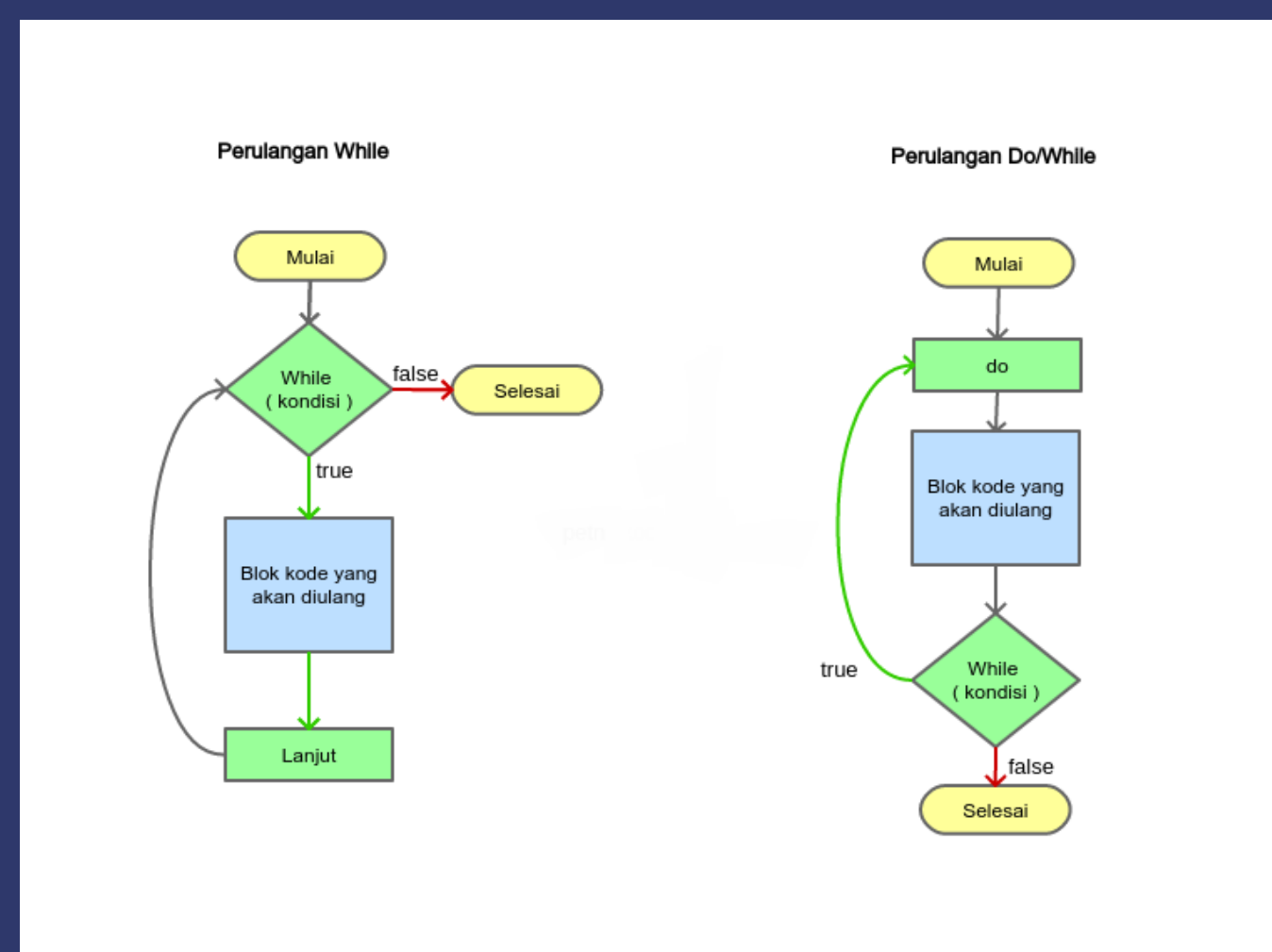
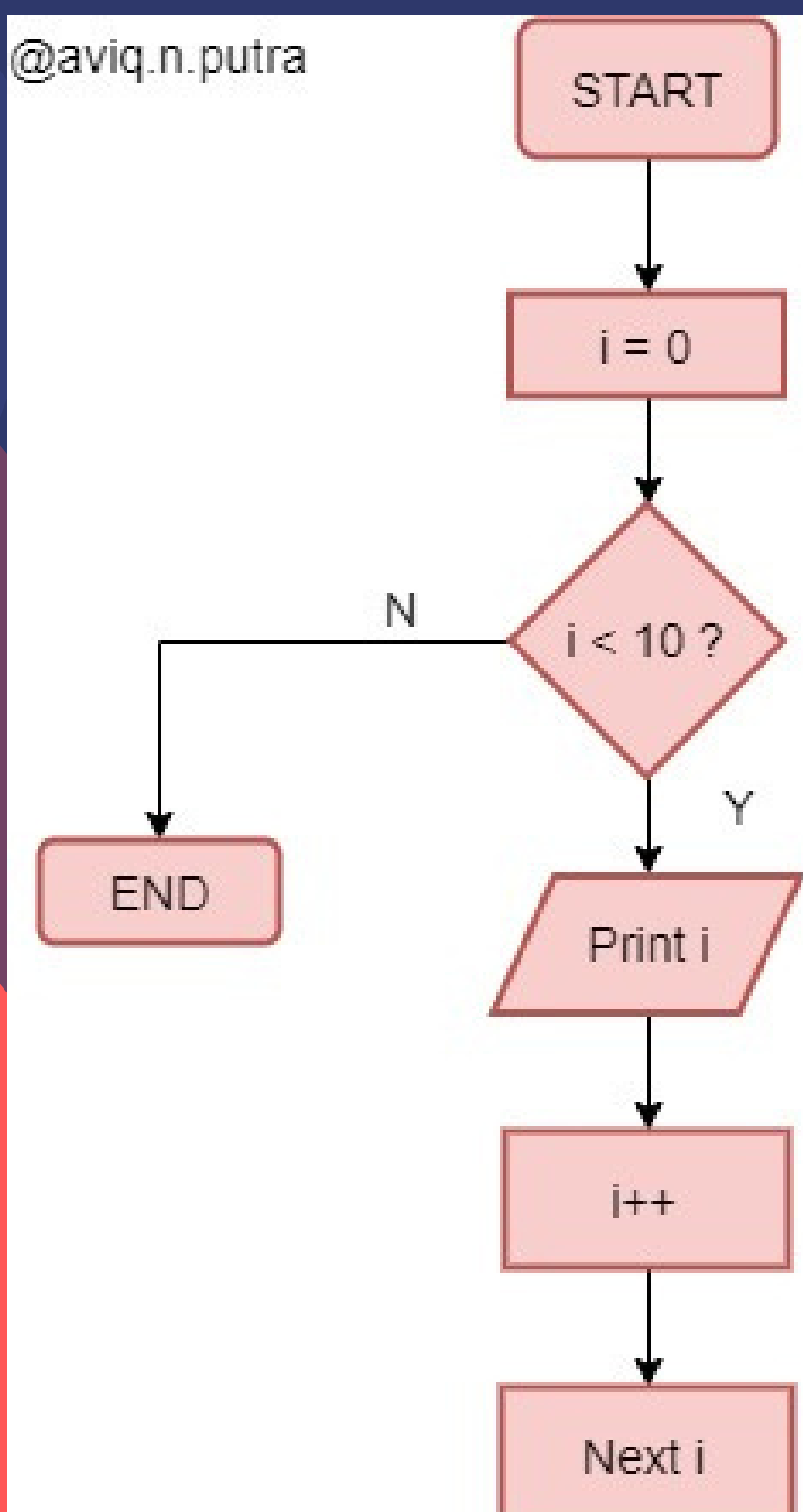
Perulangan for ini biasanya digunakan untuk perulangan yang sudah jelas perlu dilakukan berapa kali.

- Perulangan While

Perulangan While biasanya digunakan jika jumlah perulangan tidak diketahui atau memiliki kemungkinan untuk dapat dilakukan kurang dari batas perulangan

- Perulangan Do While

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



Perulangan For

Perulangan For dapat dibagi menjadi 2:

Perulangan for untuk hitung naik

Perulangan for untuk hitung turun

Perulangan for untuk hitung naik:

```
#include <iostream>

using namespace std;

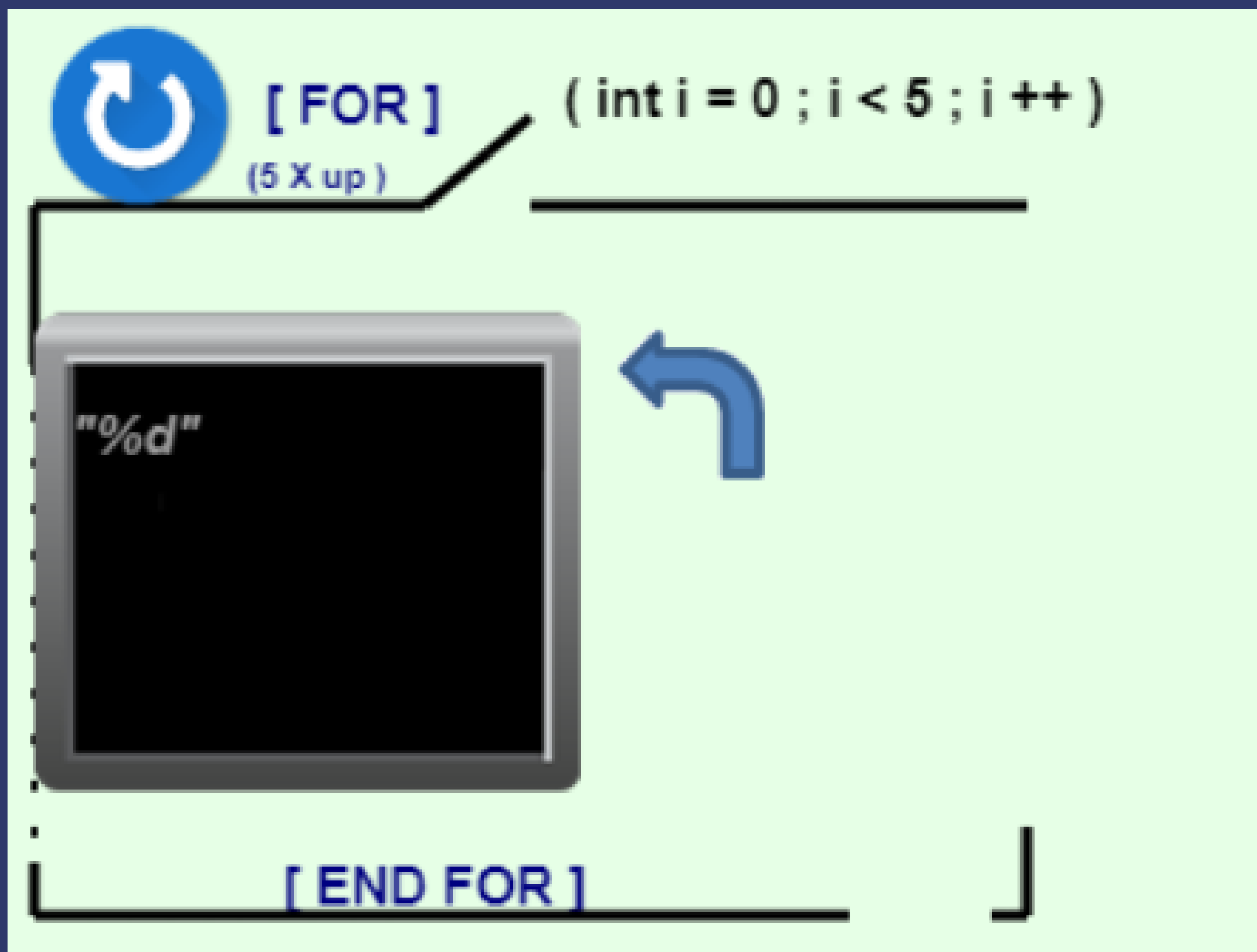
int main()
{
    for(int i = 0;i<5;i++){
        cout<<i<<endl;
    }
}
```

int i = 0 (merupakan bagian inisialisasi)

cout<<i<<endl (merupakan bagian proses)

i++ (merupakan iterasi)

i<5 (merupakan terminasi)



Maka output yang akan dicetak adalah

0
1
2
3
4
5

Hal ini disebabkan perulangan mulai dari 0 sampai lebih kecil dari 5 artinya 0 sampai satu angka lebih kecil dari 5

0
1
2
3
4
5

dan ingat kenapa ada spasi karena ada <<endl menandakan pindah ke baris baru.

Pengulangan For untuk perhitungan turun

```
9  #include <iostream>
10
11  using namespace std;
12
13  int main()
14  {
15      for(int i = 4; i >= 0; i--){
16          cout << i << endl;
17      }
18  }
```

i = 4 (merupakan inisialisasi/nilai awal)

i >= 0 (merupakan terminasi/batas atas)

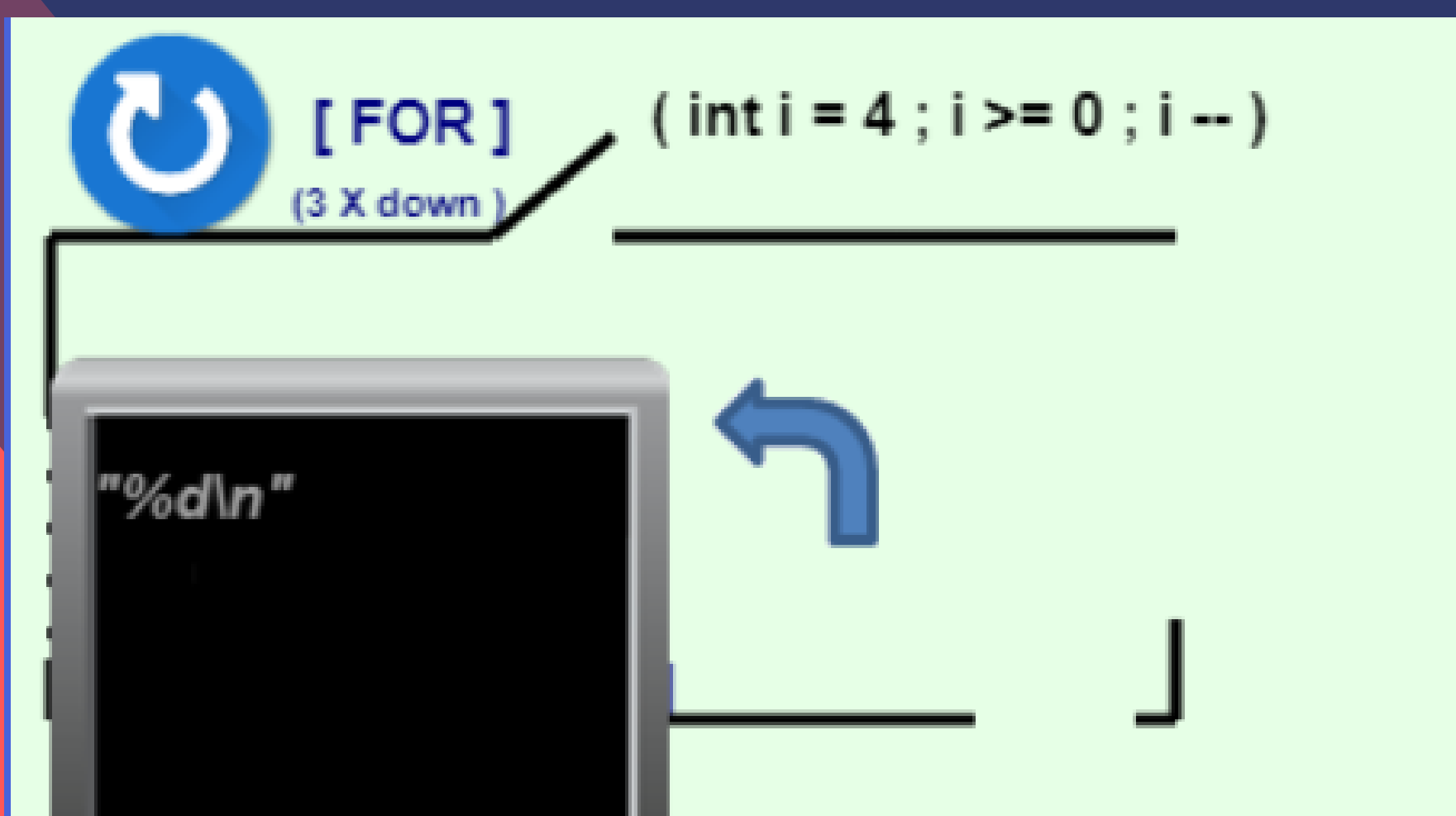
i-- (merupakan kondisi iterasi)

cout << i << endl (merupakan proses)

perlu diingat iterasi ditulis i-- karena angkanya menurun sampai 0.

Output:

4
3
2
1
0



Contoh Soal dan Penyelesaian

Miftah ingin membuat suatu aplikasi yang menerapkan prinsip perulangan yang diketahui batas awalnya dan batas akhir berupa angka dari 0 sampai 4 kira-kira dengan menggunakan perulangan apa yang paling cocok?

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: Miftah ingin membuat suatu program yang diketahui batas awal dan batas akhir (jumlah perulangannya)

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 perulangan for

Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):

1. Jika dibuat dalam algoritma kira-kira bagaimana?

Jawab:

```
9  #include <iostream>
10
11  using namespace std;
12
13  int main()
14  {
15      for(int i = 0; i < 5; i++){
16          cout << i << endl;
17      }
18  }
19
```

Contoh Soal dan Penyelesaian

Tahap 4 (Memeriksa Kembali):

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

```
9  #include <iostream>
10
11  using namespace std;
12
13  int main()
14  {
15      for(int i = 0;i<5;i++){
16          cout<<i<<endl;
17      }
18  }
19
```

Jawab: Kode tersebut tidak menghasilkan error dan hasilnya pun tepat

0
1
2
3
4
5

Daftar Pustaka

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

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