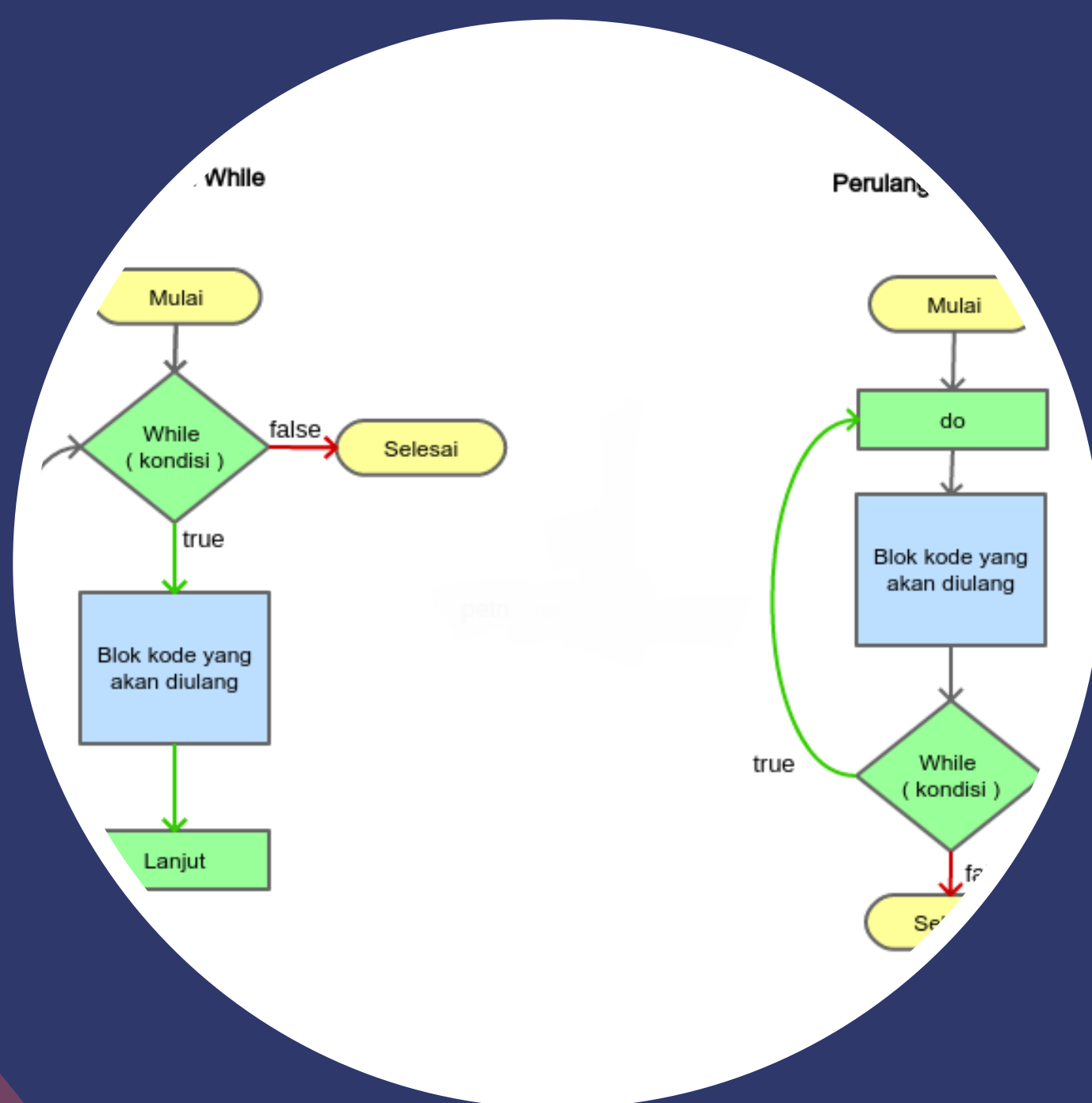


MODUL PERULANGAN

Modul Perulangan Bagian II



Disusun Oleh :
**Johannes Alexander
Putra**

Perulangan Do While

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)

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int a = 9;
7      do{
8          cout<<"genap"<<endl;
9          a--;
10     }while(a % 2 == 0);
11
12     return 0;
13
14
15 }
```



jdoodle.com/ia/Nho

Penjelasan

Konstruksi do while 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

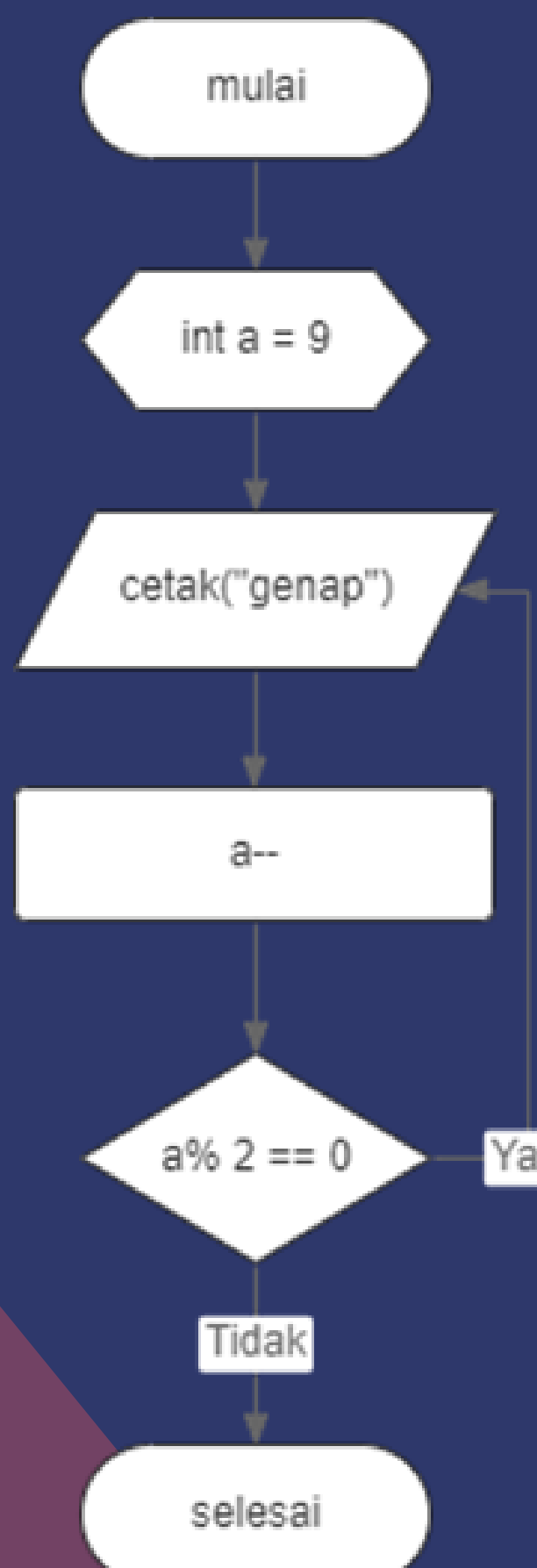
Miftah ingin membuat suatu aplikasi dalam bahasa pemrograman C++ yang menerapkan prinsip perulangan yang iterasinya berdasarkan kondisi dan tidak hanya sekadar angka awal dan angka akhir saja dan ingin mengulang minimal satu kali, Jika angka awal adalah 9. Perulangan akan mencetak “Genap” minimal satu kali dan nilai bilangan selalu berkurang 1 dan batas looping adalah ketika bilangan bernilai ganjil, penamaan variable dibebaskan, apakah informasi sudah cukup? Perulangan apa yang dipergunakan dan bagaimana code programnya.

Ada 4 Tahapan Penyelesaian masalah menurut (Polya, 2014)

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

2. Apa saja data yang diketahui?

Miftah ingin membuat aplikasi perulangan dalam bahasa C++. Iterasi perulangannya berdasarkan suatu kondisi bukan hanya sekedar angka awal dan angka akhir. Perulangan yang dilakukan minimal satu kali. Mencetak genap minimal satu kali dan berhenti ketika angka bernilai ganjil

Tahap 2 (Membuat Rencana Penyelesaian Masalah):

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

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

2. Bagaimana flowchartnya?

Jawab:

Tahap 3 (Melaksanakan Rencana Penyelesaian Masalah):
1. Bagaimana kodenya?

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int a = 9;
5      do{
6          cout<<"Genap \n";
7          a--;
8
9      }while(a % 2 == 0);
10     return 0;
11 }
```

Tahap 4 (Memeriksa Kembali):

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

Berdasarkan kode tersebut dapat disimpulkan bahwa perulangan yang dihasilkan sudah benar. Bahwa minimal diadakan perulangan minimal 1 kali.



2. Bagaimana agar perulangan seperti code tersebut mencetak “Ganjil” dan berhenti ketika angka bernilai genap, nilai bilangan selalu berkurang 1?

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int a = 9;
5      do{
6          cout<<"Ganjil \n";
7          a--;
8
9      }while(a % 2 != 0);
10     return 0;
11 }
```

Ubah kata yang dicetak dari sebelumnya genap menjadi ganjil dan ubah kondisi whilenya dari sebelumnya $a \% 2 == 0$ menjadi $a \% 2 != 0$

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