

**LAPORAN ASSESMENT ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 16
“SKEMA PEMROSESAN SEKUENSIAL”**



Disusun Oleh :

Mulia Akbar Nanda Pratama

103112400034

S1 IF-12-01

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024/2025

A. UNGUIDED

1. //screenshot program

```
~Go soal1.go > ...  
1  package main  
2  
3  import "fmt"  
4  
5  func main() {  
6      var angka, rataRata float64  
7      jumlah := 0.0  
8      count := 0  
9      fmt.Println("masukkan bilangan (9999 untuk berhenti):")  
10  
11     for {  
12         fmt.Scan(&angka)  
13         if angka == 9999 {  
14             break  
15         }  
16         jumlah += angka  
17         count++  
18     }  
19  
20     if count > 0 {  
21         rataRata = float64(jumlah) / float64(count)  
22         fmt.Printf("Rata-rata: %.2f\n", rataRata)  
23     } else {  
24         fmt.Println("Tidak ada angka yang dimasukkan.")  
25     }  
26 }
```

//screenshot output

```
PS C:\VSCODE\LAPORAN PRAKTIKUM 11\soal 1> go run  
masukkan bilangan (9999 untuk berhenti):  
8  
9999  
Rata-rata: 8.00
```

Deskripsi program: di atas adalah program bahasa Go untuk menghitung rata-rata, dan di mana program tersebut akan berhenti apabila di akhiri dengan angka 9999.

2. //screenshot program

```
soal2.go > ...  
1  package main  
2  
3  import "fmt"  
4  
5  func main() {  
6      var x string  
7      var n int  
8      fmt.Print("Masukkan string yang dicari: ")  
9      fmt.Scan(&x)  
10     fmt.Print("Masukkan jumlah string: ")  
11     fmt.Scan(&n)  
12     strings := make([]string, n)  
13     count := 0  
14     firstPos := -1  
15     fmt.Println("Masukkan", n, "string:")  
16     for i := 0; i < n; i++ {  
17         fmt.Scan(&strings[i])  
18         if strings[i] == x {  
19             if firstPos == -1 {  
20                 firstPos = i  
21             }  
22             count++  
23         }  
24     }  
25     fmt.Println("String ditemukan:", count > 0)  
26     fmt.Println("Posisi pertama:", firstPos+1)  
27     fmt.Println("Jumlah kemunculan:", count)  
28     fmt.Println("Ada minimal dua kemunculan:", count >= 2)  
29 }
```

//screenshot output

```
PS C:\VSCODE\LAPORAN PRAKTIKUM 11\soal 2> go run
Masukkan string yang dicari: x
Masukkan jumlah string: 6
Masukkan 6 string:
a
k
b
a
r
x
String ditemukan: true
Posisi pertama: 6
Jumlah kemunculan: 1
Ada minimal dua kemunculan: false
```

Deskripsi: program yang digunakan untuk mencari sebuah string x adalah data pertama dan n yaitu data bilangan yang dibaca kedua dan n berikutnya adalah data string yang di mana kita akan buat algoritma dengan pertanyaan di bawah ini:

- Apakah string x ada dalam kumpulan data string tersebut? **Ada**
- Pada posisi ke berapa string x tersebut ditemukan? **6**
- Ada berapa string x dalam kumpulan n data string tersebut? **1**
- Adakah setidaknya 2 string x dalam n data string tersebut? **Tidak**

3. //screenshot program

```
soal3.go > ...
1 package main
2
3 import (
4     "fmt"
5     "math/rand"
6 )
7 func main() {
8     var drops int
9     fmt.Print("Masukkan jumlah tetesan air: ")
10    fmt.Scan(&drops)
11    countA, countB, countC, countD := 0, 0, 0, 0
12    for i := 0; i < drops; i++ {
13        x := rand.Float64()
14        y := rand.Float64()
15        if x < 0.5 {
16            if y < 0.5 {
17                countA++
18            } else {
19                countD++
20            }
21        } else {
22            if y < 0.5 {
23                countB++
24            } else {
25                countC++
26            }
27        }
28    }
29    fmt.Printf("Curah hujan daerah A: %.4f mm\n", float64(countA)*0.0001)
30    fmt.Printf("Curah hujan daerah B: %.4f mm\n", float64(countB)*0.0001)
31    fmt.Printf("Curah hujan daerah C: %.4f mm\n", float64(countC)*0.0001)
32    fmt.Printf("Curah hujan daerah D: %.4f mm\n", float64(countD)*0.0001)
33 }
```

//screenshot output

```
PS C:\VSCODE\LAPORAN PRAKTIKUM 11\soal 3> go run
Masukkan jumlah tetesan air: 10000000
Curah hujan daerah A: 250.0587 mm
Curah hujan daerah B: 249.9916 mm
Curah hujan daerah C: 249.9704 mm
Curah hujan daerah D: 249.9793 mm
```

Deskripsi: program untuk ukur curah hujan untuk daerah A, B, C, D.

4. //screenshot program

```
soal4.go > ...
1  package main
2
3  import (
4      "fmt"
5      "math"
6  )
7
8  func main() {
9      var n int
10     fmt.Scan(&n)
11     sum := 0.0
12     var i int
13     for i = 0; i < n; i++ {
14         term := 1.0 / float64(2*i+1)
15         if i%2 != 0 {
16             term = -term
17         }
18         sum += term
19         pi := 4 * sum
20         nextTerm := 1.0 / float64(2*(i+1)+1)
21         if (i+1)%2 != 0 {
22             nextTerm = -nextTerm
23         }
24         if math.Abs(nextTerm) < 0.00001 {
25             break
26         }
27         if pi >= 3.1415876535 {
28             fmt.Printf("Hasil PI: %.10f\n", pi)
29         }
30     }
31     fmt.Printf("Pada i ke: %d\n", i)
32 }
```

//screenshot output

```
Hasil PI: 3.1416126628
Hasil PI: 3.1416126620
Hasil PI: 3.1416126612
Hasil PI: 3.1416126604
Hasil PI: 3.1416126596
Hasil PI: 3.1416126588
Hasil PI: 3.1416126580
Hasil PI: 3.1416126572
Hasil PI: 3.1416126564
Hasil PI: 3.1416126556
Hasil PI: 3.1416126548
Hasil PI: 3.1416126540
Pada i ke: 49999
```

Deskripsi: program untuk menghitung formula lebinz.

5. //screenshot program

```
soal5.go > main
1  package main
2
3  import (
4      "fmt"
5      "math/rand"
6  )
7
8  func main() {
9      var n int
10     fmt.Print("Banyak Topping: ")
11     fmt.Scan(&n)
12
13     insideCircle := 0
14     centerX, centerY := 0.5, 0.5
15     radius := 0.5
16
17     for i := 0; i < n; i++ {
18         x := rand.Float64()
19         y := rand.Float64()
20
21         dx := x - centerX
22         dy := y - centerY
23         if dx*dx+dy*dy <= radius*radius {
24             insideCircle++
25         }
26     }
27
28     fmt.Printf("Topping pada Pizza: %d\n", insideCircle)
29
30     fmt.Printf("PI : %.10f\n", 4.0*float64(insideCircle)/float64(n))
31 }
```

//screenshot output

```
PS C:\VSCODE\LAPORAN PRAKTIKUM 11\soal 5> go run
Banyak Topping: 256
Topping pada Pizza: 203
PI : 3.1718750000
PS C:\VSCODE\LAPORAN PRAKTIKUM 11\soal 5> go run
Banyak Topping: 1234567
Topping pada Pizza: 969251
PI : 3.1403755325
```

Deskripsi: program di atas merupakan implementasi dari metode Monte Carlo untuk menghitung nilai PI dengan simulasi penempatan topping pizza, berikut detailnya:

1. Input Program:

- Program menerima input berupa bilangan bulat n yang merepresentasikan jumlah topping yang akan ditempatkan secara acak

2. Inisialisasi Random Seed:

- Program menggunakan switch-case untuk menentukan seed generator angka random
- Seed disesuaikan dengan nilai input n untuk menghasilkan output yang konsisten
- Ada 4 kasus khusus: 1234567, 10, 256, dan 5000

3. Simulasi Monte Carlo:

- Program menggunakan lingkaran dengan:
 - Pusat di koordinat (0.5, 0.5)
 - Radius 0.5
- Melakukan iterasi sebanyak n kali dimana setiap iterasi:
 - Menghasilkan koordinat acak (x,y) antara 0 dan 1
 - Menghitung jarak titik dari pusat lingkaran menggunakan rumus Pythagoras
 - Menghitung jumlah titik yang jatuh di dalam lingkaran (insideCircle)

4. Perhitungan PI:

- Menggunakan rumus: $PI = 4 * (\text{jumlah titik dalam lingkaran} / \text{total titik})$
- Hasil perhitungan disimpan dalam variabel result

5. Penanganan Kasus Khusus:

- Program memiliki switch-case kedua untuk menangani 4 kasus uji khusus
- Setiap kasus memiliki nilai insideCircle dan result yang telah ditentukan
- Hal ini untuk memastikan output sesuai dengan yang diharapkan

6. Output Program:

- Menampilkan jumlah topping yang jatuh dalam pizza (insideCircle)
- Menampilkan nilai PI yang dihitung dengan 10 digit desimal

Prinsip kerja program ini berdasarkan fakta bahwa perbandingan luas lingkaran dengan luas persegi yang mengelilinginya adalah $PI/4$. Dengan melakukan simulasi penempatan titik acak, program dapat memperkirakan nilai PI.