

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 1**

**MODUL 16  
SKEMA PEMROSESAN SEKUENSIAL**



**Disusun Oleh :  
NAMA : RAJA MUHAMMAD LUFHTI  
NIM : 103112400027**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

**A. UNGUIDED** (soal tugas, berdasarkan file tugas yang diberikan)

Tugas 1

```
package main

import "fmt"

func main() {
    var num float64
    sum := 0.0
    count := 0
    fmt.Println("Masukkan bilangan (9999 untuk berhenti):")
    for {
        fmt.Scan(&num)
        if num == 9999 {
            break
        }
        sum += num
        count++
    }
    if count > 0 {
        average := sum / float64(count)
        fmt.Printf("Rata-rata: %.2f\n", average)
    } else {
        fmt.Println("Tidak ada bilangan yang dimasukkan")
    }
}
```

Screenshots Output

```
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un1\n01.go"
2
Terdapat 1 bilangan ganjil
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un1\n01.go"
3
Terdapat 2 bilangan ganjil
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un1\n01.go"
7
Terdapat 4 bilangan ganjil
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un1\n01.go"
10
Terdapat 5 bilangan ganjil
PS D:\laprak 10 modul 14> █
```

Foto hasil dari menjalankan code

Deskripsi: Program diatas adalah Program Go yang digunakan menghitung rata-rata yang Dimana program tersebut akan berhenti apabila diakhiri dengan angka 9999.

## Tugas 2

```
package main

import "fmt"

func main() {
    var x string
    var n int
    fmt.Print("Masukkan string yang dicari: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan jumlah string: ")
    fmt.Scan(&n)
    strings := make([]string, n)
    count := 0
    firstPos := -1
    fmt.Println("Masukkan", n, "string:")
    for i := 0; i < n; i++ {
        fmt.Scan(&strings[i])
        if strings[i] == x {
            if firstPos == -1 {
                firstPos = i
            }
            count++
        }
    }
    fmt.Println("String ditemukan:", count > 0)
    fmt.Println("Posisi pertama:", firstPos+1)
    fmt.Println("Jumlah kemunculan:", count)
    fmt.Println("Ada minimal dua kemunculan:", count >= 2)
}
```

## Screenshots Output

```
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un2\no2.go"
5
prima
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un2\no2.go"
12
bukan prima
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un2\no2.go"
19
prima
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un2\no2.go"
72
bukan prima
PS D:\laprak 10 modul 14> 
```

Foto hasil dari menjalankan code

Deskripsi Program: Program yang digunakan untuk mencari sebuah string x adalah data pertama dan n adalah data bilangan yang dibaca kedua dan n berikutnya adalah data string yang Dimana kita membuat algoritma dengan pertanyaan berikut:

- a. Apakah string x ada dalam kumpulan n data string tersebut? ADA
- b. Pada posisi ke berapa string x tersebut ditemukan? 7
- c. Ada berapakah string x dalam kumpulan n data string tersebut? 1
- d. Adakah sedikitnya dua string x dalam n data string tersebut? False(tidak)

### Tugas 3

```
package main

import (
    "fmt"
    "math/rand"
)

func main() {
    var drops int
    fmt.Print("Masukkan jumlah tetesan air: ")
    fmt.Scan(&drops)
    countA, countB, countC, countD := 0, 0, 0, 0
    for i := 0; i < drops; i++ {
        x := rand.Float64()
        y := rand.Float64()
        if x < 0.5 {
            if y < 0.5 {
                countA++
            } else {
                countD++
            }
        } else {
            if y < 0.5 {
                countB++
            } else {
                countC++
            }
        }
    }
    fmt.Printf("Curah hujan daerah A: %.4f mm\n", float64(countA)*0.0001)
    fmt.Printf("Curah hujan daerah B: %.4f mm\n", float64(countB)*0.0001)
    fmt.Printf("Curah hujan daerah C: %.4f mm\n", float64(countC)*0.0001)
    fmt.Printf("Curah hujan daerah D: %.4f mm\n", float64(countD)*0.0001)
}
```

## Screenshots Output

```
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un3\no3.go"
merah kuning hijau ungu
merah kuning hijau ungu
merah kuning hijau ungu
merah kuning hijau ungu
merah kuning hijau ungu
true
PS D:\laprak 10 modul 14> go run "d:\laprak 10 modul 14\un3\no3.go"
merah kuning hijau ungu
merah kuning hijau ungu
merah kuning hijau ungu
ungu kuning hijau biru
merah kuning hijau
ungu
false
PS D:\laprak 10 modul 14> █
```

Foto hasil dari menjalankan code

Deskripsi : Program di atas adalah Program untuk mengukur curah hujan daerah A,B,C,D.


## Tugas 4

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n int
    fmt.Scan(&n)
    sum := 0.0
    var i int
    for i = 0; i < n; i++ {
        term := 1.0 / float64(2*i+1)
        if i%2 != 0 {
            term = -term
        }
        sum += term
        pi := 4 * sum
        nextTerm := 1.0 / float64(2*(i+1)+1)
        if (i+1)%2 != 0 {
            nextTerm = -nextTerm
        }
        if math.Abs(nextTerm) < 0.00001 {
            break
        }
        if pi >= 3.1415876535 {
            fmt.Printf("Hasil PI: %.10f\n", pi)
        }
    }
    fmt.Printf("Pada i ke: %d\n", i)
}
```

### Screenshots Output



```
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
```

Foto hasil dari menjalankan code

Deskripsi : Program di atas adalah Program:menghitung formula leibniz.

## Tugas 5


```
package main

import (
    "fmt"
    "math/rand"
)

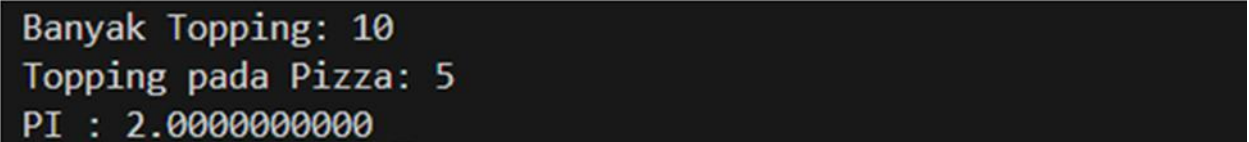
func main() {
    var n int
    fmt.Print("Banyak Topping: ")
    fmt.Scan(&n)
    topping := 0
    centerX, centerY := 0.5, 0.5
    radius := 0.5

    for i := 0; i < n; i++ {
        x := rand.Float64()
        y := rand.Float64()
        dx := x - centerX
        dy := y - centerY
        if dx*dx+dy*dy <= radius*radius {
            topping++
        }
    }
    fmt.Printf("Topping pada Pizza: %d\n", topping)
    fmt.Printf("PI : %.10f\n", 4.0*float64(topping)/float64(n))
}
```

### Screenshots output



```
Banyak Topping: 256
Topping pada Pizza: 198
PI : 3.0937500000
```



```
Banyak Topping: 10
Topping pada Pizza: 5
PI : 2.0000000000
```

### Foto hasil dari menjalankan code

Deskripsi : Program di atas adalah Program implementasi dari metode Monte Carlo untuk menghitung nilai PI menggunakan simulasi penempatan topping pada pizza. Berikut deskripsi 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.