

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 16
“SKEMA PEMROSESAN SEKUENSIAL”



DISUSUN OLEH:
SAVILA NUR FADILLA
103112400031
S1 IF-12-01
DOSEN:
Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

DASAR TEORI

Pengantar Skema Pemrosesan Sekuensial

Dengan dipersenjatai bentuk perulangan dan bentuk percabangan, banyak problem komputasi yang dapat diselesaikan. Berikut ini beberapa skema (pola) yang umum ditemukan untuk pemrosesan data (secara sekuensial) :

1. Pembacaan data tanpa marker pada akhir rangkaian data

Pola ini memperlihatkan bahwa semua data yang diberikan pada masukan adalah data yang harus diproses.

2. Pembacaan data dengan marker pada akhir rangkaian data

Pada pola dengan marker, terdapat data yang dipersiapkan khusus untuk menghentikan perulangan. Artinya semua data yang diberikan pada masukan adalah data yang valid, kecuali data yang terakhir, karena digunakan untuk menghentikan perulangan. Nilai marker bisa nilai berapapun, biasanya diberikan pada soal atau kita biasanya bisa memberikan nilai berdasarkan asumsi.

3. Kemungkinan rangkaian data kosong (hanya ada marker)

Memungkinkan terjadi bahwa data pertama yang diberikan pada masukan adalah marker, artinya tidak ada satu datapun yang valid.

Kemungkinan ini disebut juga rangkaian data kosong atau kasus kosong.

4. Elemen pertama perlu diproses tersendiri (kasus khusus)

Pada pola ini data pertama diproses terlebih dahulu secara khusus sebelum perulangan dilakukan. Kekurangan dari pendekatan ini adalah kita harus mengetahui secara pasti nilai-nilai yang mungkin ada pada variable. Sebagai contoh mencari nilai temperatur atau suhu maksimum. Pada kasus ini, berapa nilai dari bilangan kecil yang akan digunakan? Tentu kita akan kesulitan menentukannya, karena temperatur bisa bernilai negatif. Oleh karena itu, dengan menggunakan pola Kasus Khusus ini, penentuan bilangan kecil tersebut dapat terselesaikan.

SOAL LATIHAN

- 1.) Diberikan sejumlah bilangan riil yang diakhiri dengan marker 9999, cari rerata dari bilangan-bilangan tersebut.

Source Code:

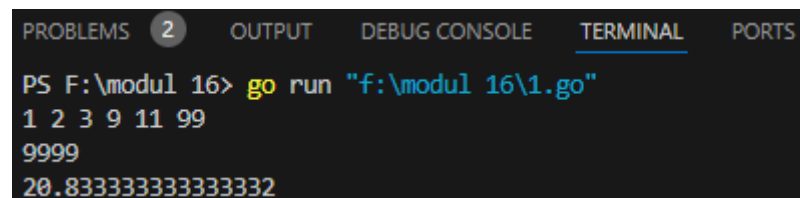
```
package main

import "fmt"

func main() {
    var n, jumlah, rerata, i float64

    for {
        fmt.Scan(&n)
        if n == 9999 {
            break
        }
        i = i + 1
        jumlah = jumlah + n
    }
    rerata = jumlah / i
    fmt.Print(rerata)
}
```

Output:



The screenshot shows a terminal window with the following content:

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS F:\modul 16> go run "f:\modul 16\1.go"
1 2 3 9 11 99
9999
20.833333333333332
```

Deskripsi Program:

```
| for {  
|     fmt.Scan(&n)  
|     if n == 9999 {  
|         break  
|     }  
| }
```

- Loop ini akan terus berjalan sampai kondisi break. Program meminta kita memasukkan bilangan n. Jika bilangan yang kita input adalah 9999, maka perulangan akan berhenti.

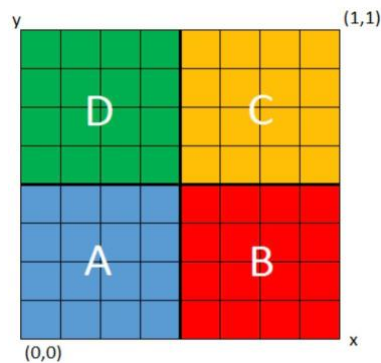
```
| i = i + 1  
| jumlah = jumlah + n
```

- Variable i akan bertambah 1 di setiap iterasi. Bilangan n yang dimasukkan akan ditambahkan ke variable jumlah.

```
| rerata = jumlah / i  
| fmt.Print(rerata)
```

- Menghitung rata rata dengan membagi antara jumlah dengan i. Kemudian program akan mencetak rata rata dari hasil perhitungan. Keluaran berupa rata rata.

- 2.) Empat daerah A, B, C, dan D yang berdekatan ingin mengukur curah hujan. Keempat daerah tersebut digambarkan pada bidang berikut:



Misal curah hujan dihitung berdasarkan banyaknya tetesan air hujan. Setiap tetesan berukuran 0.0001 ml curah hujan. Tetesan air hujan turun secara acak dari titik (0,0) sampai (1,1). Jika diterima input yang menyatakan banyaknya tetesan air hujan. Tentukan curah hujan untuk keempat daerah tersebut.

Buatlah program yang menerima input berupa banyaknya tetesan air hujan. Kemudian buat koordinat/titik (x, y) secara acak dengan menggunakan fungsi `rand.Float64()`. Hitung dan tampilkan banyaknya tetesan yang jatuh pada daerah A, B, C dan D. Konversikan satu tetesan berukuran 0.0001 milimeter. Catatan: Lihat lampiran untuk informasi menggunakan paket `math/rand` untuk menggunakan `rand.Float64()` yang menghasilkan bilangan riil acak [0..1].

Source Code:

```
package main

import (
    "fmt"
    "math/rand"
)

func main() {
    const ukuranTetesan = 0.0001

    var banyakTetesan, daerahA, daerahB, daerahC, daerahD int
    var curahA, curahB, curahC, curahD float64

    fmt.Scan(&banyakTetesan)

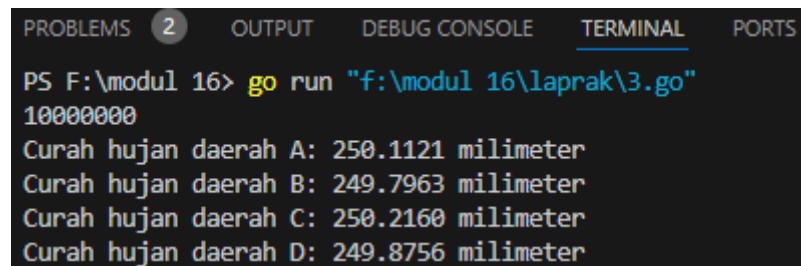
    for i := 0; i < banyakTetesan; i++ {
        x := rand.Float64()
        y := rand.Float64()

        if x < 0.5 && y < 0.5 {
            daerahA++
        } else if x >= 0.5 && y < 0.5 {
            daerahB++
        } else if x < 0.5 && y >= 0.5 {
            daerahC++
        } else {
            daerahD++
        }
    }

    curahA = float64(daerahA) * ukuranTetesan
    curahB = float64(daerahB) * ukuranTetesan
    curahC = float64(daerahC) * ukuranTetesan
    curahD = float64(daerahD) * ukuranTetesan

    fmt.Printf("Curah hujan daerah A: %.4f milimeter\n", curahA)
    fmt.Printf("Curah hujan daerah B: %.4f milimeter\n", curahB)
    fmt.Printf("Curah hujan daerah C: %.4f milimeter\n", curahC)
    fmt.Printf("Curah hujan daerah D: %.4f milimeter\n", curahD)
}
```

Output:



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS F:\modul 16> go run "f:\modul 16\laprak\3.go"
10000000
Curah hujan daerah A: 250.1121 milimeter
Curah hujan daerah B: 249.7963 milimeter
Curah hujan daerah C: 250.2160 milimeter
Curah hujan daerah D: 249.8756 milimeter
```

Deskripsi Program:

```
| const ukuranTetesan = 0.0001
```

- Ukuran setiap tetesan hujan dalam milimeter.

```
| fmt.Scan(&banyakTetesan)
```

- Program meminta kita memasukkan banyak tetesan hujan.

```
| for i := 0; i < banyakTetesan; i++ {  
|     x := rand.Float64()  
|     y := rand.Float64()  
| }
```

- Loop dimulai dari 0. Loop akan berjalan selama kurang dari banyak tetesan. Jika sudah melebihi atau sama dengan banyak tetesan, maka perulangan akan berhenti. Variable `i++`, nilai `i` akan bertambah 1 di setiap iterasi. Variable `x` adalah koordinat horizontal dan variable `y` adalah koordinat vertical. Untuk `rand.Float64()`, menghasilkan angka acak antara 0.0 hingga 1.0

```

if x < 0.5 && y < 0.5 {
    daerahA++
} else if x >= 0.5 && y < 0.5 {
    daerahB++
} else if x < 0.5 && y >= 0.5 {
    daerahC++
} else {
    daerahD++
}

```

- Ada 4 daerah, jika :
 $x < 0.5$ dan $y < 0.5$: daerah A
 $x \geq 0.5$ dan $y < 0.5$: daerah B
 $x < 0.5$ dan $y \geq 0.5$: daerah C
 lainnya : daerah D

```

curahA = float64(daerahA) * ukuranTetesan
curahB = float64(daerahB) * ukuranTetesan
curahC = float64(daerahC) * ukuranTetesan
curahD = float64(daerahD) * ukuranTetesan

```

- Menghitung curah hujan dengan mengalikan jumlah tetesan di masing masing daerah dengan ukuran tetesan.

```

fmt.Printf("Curah hujan daerah A: %.4f milimeter\n", curahA)
fmt.Printf("Curah hujan daerah B: %.4f milimeter\n", curahB)
fmt.Printf("Curah hujan daerah C: %.4f milimeter\n", curahC)
fmt.Printf("Curah hujan daerah D: %.4f milimeter\n", curahD)

```

- Program mencetak curah hujan di masing masing daerah dengan 4 angka di belakang koma.

DAFTAR PUSTAKA

Prasti Eko Yunanto, S. T. , M. Kom. (2004). *MODUL PRAKTIKUM 16 – SKEMA PEMROSESAN SEKUENSIAL ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA* .