

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



DISUSUN OLEH:
DWI OKTA SURYANINGRUM
103112400066
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

Komposisi Definisi : Pemrosesan sekuensial adalah pemrosesan secara satu-persatu, dari sekumpulan informasi sejenis yang setiap elemennya dapat diakses dengan keterurutan tertentu (ada suksesor), jadi seakan-akan kumpulan elemen merupakan "Deret" elemen. Elemen yang akan diproses dapat bertipe dasar (integer, real, character, boolean), tetapi dapat juga bertipe komposisi (misalnya Point).

Deret Elemen dapat merupakan elemen yang dibaca satu per satu dari input device, nilai elemen suatu tabel atau matriks, disimpan dalam media penyimpanan sekunder (file), atau merupakan elemen list, dsb. Kumpulan informasi itu disimpan sedemikian rupa, sehingga selalu dikenali melalui primitif yang mampu untuk memberikan: - Elemen pertama (First_Elmt) - Elemen yang siap diproses (Current_Elmt) - Elemen yang diakses setelah Current_Elmt (Next_Elmt) - Tanda akhir proses EOP Skema sekuensial ini dibangun untuk mendapatkan suatu pola umum, sebab instruksi pengulangan seperti yang telah dibahas di bab sebelumnya hanya berfokus kepada bentuk pengulangan, belum mencakup semua abstraksi yang dibutuhkan untuk suatu proses sekuensial.

Spesifikasi primitif : Primitif untuk skema sekuensial diterjemahkan menjadi prosedur dan nilai variabel sebagai berikut

1. procedure First_Elmt { Aksi yang memberikan elemen pertama yang akan diproses secara sekuensial. I.S. : sembarang F.S : Current_Elmt berisi sebuah elemen yang akan diproses }
2. procedure Next_Elmt { Aksi yang memberikan elemen berikutnya dari Current_Elmt sesuai dengan aturan akses dari pengaturan deret elemen : - untuk elemen yang disimpan secara kontigu (array), akses ke elemen berikutnya dapat dilakukan melalui indeks : indeks dari next element adalah suksesor dari Current-element. - untuk elemen yang disimpan dalam media sekunder, akses ke elemen berikutnya dilakukan lewat suatu aksi yang tersedia (Contoh : mesin karakter) - untuk elemen yang diatur sesuai dengan aturan lain, harus ditentukan fungsi yang memungkinkan teraksesnya elemen yang berikutnya. Realisasi dari aksi ini tergantung dari bagaimana informasi itu disimpan dalam memori : I.S. : Current_Elmt F.S : Current_Elmt = Next_Elmt (Current-Elmt) }
3. EOP: boolean { Berharga true jika proses sekuensial harus dihentikan. Dapat diimplementasi sebagai fungsi element Bagaimana EOP ini berharga true ? Ada dua

macam model : 1. Model dengan MARK. Elemen terakhir adalah elemen "fiktif", sebetulnya bukan anggota elemen yang diproses, informasinya lain dari pada elemen yang diproses secara lumrah. Model dengan Mark justru lebih banyak dipakai dalam pemrograman praktis. 2. Model tanpa MARK : elemen terakhir mengandung informasi yang memberitahukan bahwa elemen tsb. adalah elemen yang terakhir }

SOAL LATIHAN

Statement Percabangan

1. Latihan Soal Nomor 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 number, sum float64
    var count int

    for {
        fmt.Scan(&number)

        if number == 9999 {
            break
        }

        sum += number
        count++
    }

    if count == 0 {
        fmt.Println("Tidak ada bilangan untuk dihitung rata-ratanya.")
    } else {
        average := sum / float64(count)
        fmt.Printf("Rata-rata : %.2f\n", average)
    }
}
```

Output:

```
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul_16/latsol1.go"
1.2
2.3
4.7
9999
Rata-rata : 2.73
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul_16/latsol1.go"
9999
Tidak ada bilangan untuk dihitung rata-ratanya.
○ mymac@192 praktikum1 %
```

2. Latihan Soal nomor 3

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() {
    rand.Seed(1)

    var drops int
    fmt.Scan(&drops)

    const dropSize = 0.0001 // Ukuran tiap tetesan dalam ml
    countA, countB, countC, countD := 0, 0, 0, 0

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

        if x < 0.5 && y >= 0.5 {
            countA++
        } else if x >= 0.5 && y >= 0.5 {
            countB++
        } else if x < 0.5 && y < 0.5 {
            countC++
        } else if x >= 0.5 && y < 0.5 {
            countD++
        }
    }
}
```

```

}

// Hitung curah hujan
rainA := float64(countA) * dropSize
rainB := float64(countB) * dropSize
rainC := float64(countC) * dropSize
rainD := float64(countD) * dropSize

// Output hasil
fmt.Printf("Curah hujan daerah A: %.4f milimeter\n", rainA)
fmt.Printf("Curah hujan daerah B: %.4f milimeter\n", rainB)
fmt.Printf("Curah hujan daerah C: %.4f milimeter\n", rainC)
fmt.Printf("Curah hujan daerah D: %.4f milimeter\n", rainD)
}

```

Output:

```

● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul_16/latsol2.go"
10000000
Curah hujan daerah A: 250.1511 milimeter
Curah hujan daerah B: 250.1277 milimeter
Curah hujan daerah C: 249.8925 milimeter
Curah hujan daerah D: 249.8287 milimeter

```

DAFTAR PUSTAKA

(N.d.). Retrieved from https://mgmptiksemarang.wordpress.com/wp-content/uploads/2011/08/10-skemaseq_nopw.pdf