

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



Disusun oleh :

NAMA : Felix Pedrosa Valentino

NIM : 103112400056

S1 IF – 12 - 01

Dosen Pengampu :

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).

Pembacaan Data Tanpa Marker pada Akhir Rangkaian Data

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

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.

Kemungkinan Rangkaian Data Kosong, Hanya Ada Marker

Pola dengan 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.

Elemen Pertama Perlu Diproses Tersendiri / Kasus Khusus

Pada pola ini data pertama diproses terlebih dahulu secara khusus sebelum perulangan dilakukan. Apabila melihat contoh pencarian nilai maksimum di atas, terlihat bahwa nilai variabel max selalu diinisialisasi oleh sebuah nilai BILANGAN KECIL berapapun. Kekurangan dari pendekatan ini adalah kita harus mengetahui secara pasti nilai-nilai yang mungkin ada pada variabel dat, yang mana nilai pada variabel dat tersebut TIDAK BOLEH lebih kecil dibandingkan nilai dari BILANGAN KECIL yang digunakan saat inisialisasi.

LATIHAN SOAL

1.) Latihan Soal 1

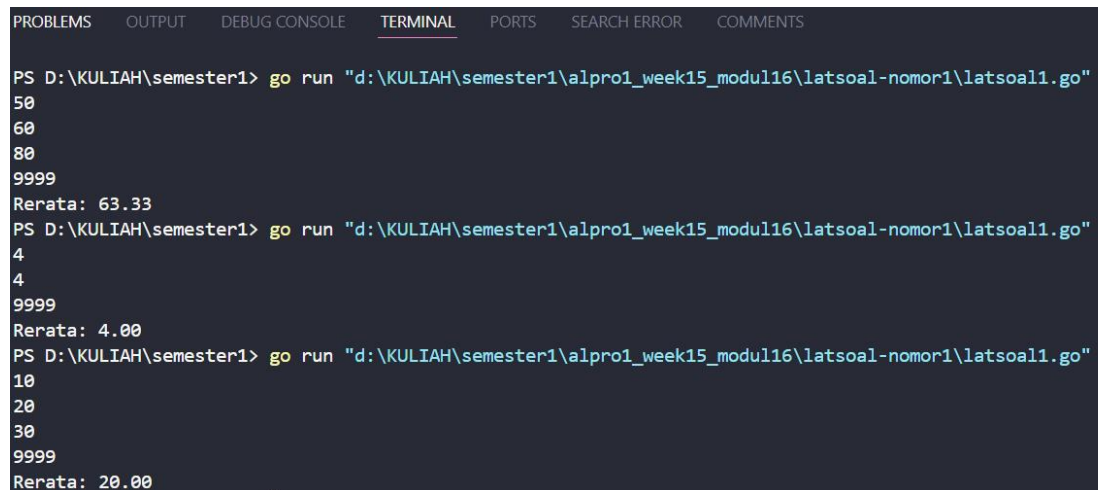
Source Code :

```
package main

import "fmt"

func main() {
    var num, jumlah, count float64
    for {
        fmt.Scan(&num)
        if num == 9999 {
            break
        }
        jumlah += num
        count++
    }
    if count > 0 {
        fmt.Printf("Rerata: %.2f\n", jumlah/count)
    } else {
        fmt.Println("Tidak ada bilangan yang dimasukkan.")
    }
}
```

Output :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS

PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week15_modul16\latsoal-nomor1\latsoal1.go"
50
60
80
9999
Rerata: 63.33
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week15_modul16\latsoal-nomor1\latsoal1.go"
4
4
9999
Rerata: 4.00
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week15_modul16\latsoal-nomor1\latsoal1.go"
10
20
30
9999
Rerata: 20.00
```

Deskripsi Program :

Program di atas berfungsi untuk menghitung rata-rata dari serangkaian angka yang dimasukkan oleh pengguna. Proses dimulai dengan mendeklarasikan variabel `num`, `jumlah`, `count` bertipe `float64`, `num` untuk menyimpan angka yang dimasukkan, `jumlah` untuk menjumlahkan semua angka, dan `count` untuk menghitung jumlah angka yang dimasukkan. Program kemudian memasuki loop tak terbatas yang akan terus meminta input dari pengguna. Jika pengguna memasukkan angka 9999, program akan menghentikan proses input. Setiap angka yang dimasukkan akan ditambahkan ke variabel `jumlah`, dan variabel `count` akan bertambah satu. Setelah pengguna selesai memasukkan angka, program akan memeriksa apakah ada angka yang dimasukkan. Jika ada, program akan menghitung dan menampilkan rata-rata dari angka-angka tersebut dengan format dua desimal. Jika tidak ada angka yang dimasukkan, program akan menampilkan pesan "Tidak ada bilangan yang dimasukkan."

2.) Latihan Soal 2

Source Code :

```
package main

import "fmt"

func main() {
    var x string
    var n int
    fmt.Scan(&x)
    fmt.Scan(&n)
    strings := make([]string, n)
    for i := 0; i < n; i++ {
        fmt.Scan(&strings[i])
    }
    // a. Apakah string x ada dalam kumpulan data?
    ditemukan := false
    for _, s := range strings {
        if s == x {
            ditemukan = true
            break
        }
    }
}
```

```

    }
}
if ditemukan {
    fmt.Println("String ditemukan.")
} else {
    fmt.Println("String tidak ditemukan.")
}
// b. Pada posisi ke berapa string x ditemukan?
posisi := -1
for i, s := range strings {
    if s == x {
        posisi = i + 1 // posisi mulai dari 1
        break
    }
}
if posisi != -1 {
    fmt.Printf("String ditemukan pada posisi ke-%d.\n", posisi)
} else {
    fmt.Println("String tidak ditemukan pada posisi manapun.")
}
// c. Ada berapakah string x dalam kumpulan data?
count := 0
for _, s := range strings {
    if s == x {
        count++
    }
}
fmt.Printf("Jumlah string x: %d\n", count)
// d. Adakah sedikitnya dua string x dalam kumpulan data?
if count >= 2 {
    fmt.Println("Ada sedikitnya dua string x dalam kumpulan data.")
} else {
    fmt.Println("Tidak ada sedikitnya dua string x dalam kumpulan data.")
}
}

```

Output :

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week15_modul16\latsoal-nomor2\latsoal2.go"
jambu
4
jeruk
nanas
mangga
melon
String tidak ditemukan.
String tidak ditemukan pada posisi manapun.
Jumlah string x: 0
Tidak ada sedikitnya dua string x dalam kumpulan data.
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week15_modul16\latsoal-nomor2\latsoal2.go"
jambu
4
jambu
jambu
nanas
durian
String ditemukan.
String ditemukan pada posisi ke-1.
Jumlah string x: 2
Ada sedikitnya dua string x dalam kumpulan data.
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week15_modul16\latsoal-nomor2\latsoal2.go"
nanas
3
nanas
nanas
nanas
String ditemukan.
String ditemukan pada posisi ke-1.
Jumlah string x: 3
Ada sedikitnya dua string x dalam kumpulan data.

```

Deskripsi Program :

Program di atas dirancang untuk memeriksa keberadaan dan frekuensi suatu string dalam kumpulan data string yang dimasukkan oleh pengguna. Pertama, program meminta pengguna untuk memasukkan sebuah string x dan jumlah n yang menunjukkan berapa banyak string yang akan dimasukkan. Setelah itu, program membuat sebuah slice bernama strings dengan panjang n untuk menyimpan string-string tersebut. Pengguna kemudian diminta untuk memasukkan n string, yang disimpan dalam slice tersebut.

Setelah semua string dimasukkan, program melakukan beberapa pemeriksaan: pertama, ia mencari apakah string x ada dalam kumpulan data dan memberikan umpan balik kepada pengguna. Jika ditemukan, program akan mencetak pesan bahwa string tersebut ditemukan. Selanjutnya, program mencari posisi pertama di mana string x ditemukan dalam kumpulan data dan menampilkan posisi tersebut (dengan indeks mulai dari 1). Kemudian, program menghitung jumlah kemunculan string x dalam kumpulan data dan menampilkannya. Terakhir, program

memeriksa apakah ada sedikitnya dua kemunculan string x dalam kumpulan data dan memberikan informasi yang sesuai.

3.) Latihan Soal 3

Source Code :

```
package main

import (
    "fmt"
    "math/rand"
)

func main() {
    const ukuranTetesan = 0.0001

    var jumlahTetesan, daerahA, daerahB, daerahC, daerahD int
    var curahHujanA, curahHujanB, curahHujanC, curahHujanD float64

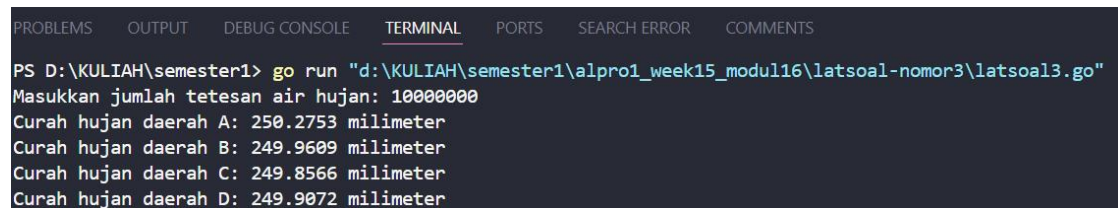
    fmt.Print("Masukkan jumlah tetesan air hujan: ")
    fmt.Scan(&jumlahTetesan)

    for i := 0; i < jumlahTetesan; 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++
        }
    }

    curahHujanA = float64(daerahA) * ukuranTetesan
    curahHujanB = float64(daerahB) * ukuranTetesan
    curahHujanC = float64(daerahC) * ukuranTetesan
    curahHujanD = float64(daerahD) * ukuranTetesan
}
```

```
fmt.Printf("Curah hujan daerah A: %.4f milimeter\n", curahHujanA)
fmt.Printf("Curah hujan daerah B: %.4f milimeter\n", curahHujanB)
fmt.Printf("Curah hujan daerah C: %.4f milimeter\n", curahHujanC)
fmt.Printf("Curah hujan daerah D: %.4f milimeter\n", curahHujanD)
}
```

Output :



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week15_modul16\latsoal-nomor3\latsoal3.go"
Masukkan jumlah tetesan air hujan: 10000000
Curah hujan daerah A: 250.2753 milimeter
Curah hujan daerah B: 249.9609 milimeter
Curah hujan daerah C: 249.8566 milimeter
Curah hujan daerah D: 249.9072 milimeter
```

Deskripsi Program :

Program di atas berfungsi untuk mensimulasikan distribusi curah hujan di empat daerah berdasarkan jumlah tetesan air hujan yang dimasukkan oleh pengguna. Program dimulai dengan mendeklarasikan sebuah konstanta ukuranTetesan yang merepresentasikan ukuran setiap tetesan air hujan, yaitu 0.0001 milimeter. Pengguna diminta untuk memasukkan jumlah tetesan air hujan yang ingin disimulasikan. Selanjutnya, program menggunakan loop untuk menghasilkan koordinat acak (x, y) untuk setiap tetesan, yang kemudian digunakan untuk menentukan daerah mana tetesan tersebut jatuh. Daerah dibagi menjadi empat kuadran: daerah A ($x \leq 0.5$ dan $y \leq 0.5$), daerah B ($x > 0.5$ dan $y \leq 0.5$), daerah C ($x \leq 0.5$ dan $y > 0.5$), dan daerah D ($x > 0.5$ dan $y > 0.5$). Setiap kali tetesan jatuh ke salah satu daerah, penghitung untuk daerah tersebut akan bertambah satu. Setelah semua tetesan diproses, program menghitung curah hujan untuk masing-masing daerah dengan mengalikan jumlah tetesan yang jatuh di daerah tersebut dengan ukuran tetesan. Akhirnya, program menampilkan curah hujan yang dihitung untuk setiap daerah dalam format milimeter dengan empat angka desimal.

DAFTAR PUSTAKA

Prasti Eko Yunanto, S.T., M.Kom. (2024). MODUL PRAKTIKUM 16 –
SKEMA PEMROSESAN SEKUENSIAL
ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA (MODUL 16)