

TP Modul ke-7 (ARRAY)

Nama: Muhamad Alwan Suryadi

NIM: 103032400104

Kelas: IT-48-01

Soal 1

```
package main
import "fmt"

func main() {
    var n int
    const NMAX = 10
    fmt.Scan(&n)
    var A [NMAX]string
    var i int
    for i = 0; i < n; i++ {
        fmt.Scan(&A[i])
    }
    fmt.Println()
    for i = 0; i < n; i++ {
        fmt.Println(A[i])
    }
}
```

```
D:\Matkul smester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal1.go
5
Telkom
University
sinergi
bangun
negeri

Telkom
University
sinergi
bangun
negeri

D:\Matkul smester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal1.go
1
telyutizen

telyutizen

D:\Matkul smester 2\Algoritma Pemrograman\praktikum>
```

Soal 2

```
package main
import "fmt"

const NMAX int = 10
type tabInt [NMAX]int

func bacaData(A *tabInt, n *int) {
    fmt.Scan(n)
    if *n > NMAX {
        *n = NMAX
    }
    for i := 0; i < *n; i++ {
        fmt.Scan(&A[i])
    }
}

func cetakData(A tabInt, n int) {
    if n == 0 {
        fmt.Println("Array kosong")
    }
    for i := 0; i < n; i++ {
        fmt.Print(A[i], " ")
    }
    fmt.Println()
}

func main() {
    var data tabInt
    var banyakData int
    bacaData(&data, &banyakData)
    fmt.Println()
    cetakData(data, banyakData)
}
```

```
D:\Matkul smester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal2.go
5
11 23 34 45 67

11 23 34 45 67

D:\Matkul smester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal2.go
11
45 76 34 28 12 45 76 34 28 12 33

45 76 34 28 12 45 76 34 28 12

D:\Matkul smester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal2.go
0

Array kosong

D:\Matkul smester 2\Algoritma Pemrograman\praktikum>
```

Soal 3

```
package main

import "fmt"

const NMAX int = 5

type tabInt struct {
    info [NMAX]int
    n    int
}

func bacaData(A *tabInt) {
    var x int
    if A.n < NMAX {
        fmt.Scan(&x)
        A.info[A.n] = x
        A.n++
    }
}

func cetakData(A tabInt) {
    if A.n > 0 {
        for i := 0; i < A.n; i++ {
            fmt.Print(A.info[i], " ")
        }
        fmt.Println()
    }
}

func main() {
    var data tabInt
    bacaData(&data)
    bacaData(&data)
    bacaData(&data)
    bacaData(&data)
    bacaData(&data)
    bacaData(&data)
    bacaData(&data)
    cetakData(data)
}
```

```
D:\Matkul smester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal3.go
11 22 33 44 55 66
11 22 33 44 55

D:\Matkul smester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal3.go
4 0 2 5 7 1 4
4 0 2 5 7

D:\Matkul smester 2\Algoritma Pemrograman\praktikum>
```

Disclaimer:

Saya menemukan kejanggalan dalam soal ini, dalam variable NMAX di deklarasi berupa 5, maka artinya array tidak akan menampung data lebih dari 5 karena akan menghasilkan error. Tetapi dalam test case pada soal dibawah ini.

No	Masukan	Keluaran
1	11 22 33 44 55 66	11 22 33 44 55
2	4 0 2 5 7 1 4	4 0 2 5 7 1

Menurut saya terdapat kesalahan karena test case kedua menghasilkan array dengan jumlah data sebanyak 6 yaitu 4,0,2,5,7,1 yang mana

seharusnya program terhenti dan tercetak di angka 7 karena merupakan batas jumlah data yang sudah dideklarasikan. Maka menurut sepemahaman saya test case yang benar adalah test case yang saya hasilkan.

Soal 4

```

package main

import "fmt"

const MAX int = 10

type tabInt [MAX]int

func main() {
    var data tabInt
    var nData int

    baca(&data, &nData)
    cetak(data, nData)
    fmt.Println(jumlah(data, nData))
    fmt.Println(rataRata(data, nData))
}

func baca(A *tabInt, n *int) {
    var num int
    var stop bool
    *n = 0
    stop = true

    for stop && *n < MAX {
        fmt.Scan(&num)

        if num == 0 {
            stop = false
        } else {
            if num < 0 {
                num = -num
            }
            A[*n] = num
            *n++
        }
    }
}

func cetak(A tabInt, n int) {
    if n > 0 {
        for i := 0; i < n; i++ {
            fmt.Print(A[i], " ")
        }
        fmt.Println()
    } else {
        fmt.Println("Array kosong")
    }
}

func jumlah(A tabInt, n int) int {
    total := 0
    for i := 0; i < n; i++ {
        total += A[i]
    }
    return total
}

func rataRata(A tabInt, n int) float64 {
    if n > 0 {
        return float64(jumlah(A, n)) / float64(n)
    }
    return 0.0
}

```

```

D:\Matkul semester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal4.go
10 2 23 -34 12 0
10 2 23 34 12
81
16.2

```

```

D:\Matkul semester 2\Algoritma Pemrograman\praktikum>go run TP_MOD_07_soal4.go
1 2 3 -4 5 6 7 -8 9 10 -11 0
1 2 3 4 5 6 7 8 9 10
55
5.5

```

```

D:\Matkul semester 2\Algoritma Pemrograman\praktikum>

```

Disclaimer:

algoritma

baca(data)

cetak(data)

output(penjumlahan(data, nData))

output(rataRata(data, nData))

endprogram

Perlu diperhatikan bahwa dalam implementasi ini, terdapat ketidaksesuaian antara cara pemanggilan prosedur di program utama dengan spesifikasi parameter yang seharusnya. Berdasarkan deklarasi `tabInt` sebagai alias `array[NMAX]` (bukan struct yang menyimpan jumlah data seperti no. 3), maka:

1. Prosedur `baca` seharusnya menerima 2 parameter:
 - Parameter array untuk menyimpan data (`A *tabInt`)
 - Parameter jumlah data (`n *int`)
2. Prosedur `cetak` seharusnya menerima 2 parameter:
 - Parameter array berisi data (`A tabInt`)
 - Parameter jumlah data (`n int`)

Pemanggilan dengan 1 parameter (`data` saja) tidak cukup karena informasi jumlah data (`n`) tidak terkandung dalam tipe `tabInt` yang didefinisikan sebagai array biasa. Maka dari itu saya mengubah cara pemanggilan prosedur `baca` dan `cetak` menjadi 2 parameter, yaitu `data` sebagai penunjang untuk parameter array dan `nData` untuk jumlah data arraynya.

Terlebih dalam contoh deklarasi prosedur `baca` dan `cetak` saja diberikan contoh dengan menggunakan 2 parameter

procedure baca(in/out A : tabInt, in/out n : integer)

{ IS: Array bilangan bulat (A) dan banyak elemen (n) terdefinisi sembarang.

Proses: Setiap bilangan yang terbaca ditampung dalam sebuah

procedure cetak(in A : tabInt, in n : integer)

{ IS: Array bilangan bulat (A) dan banyak elemen (n) terdefinisi

Karena salah satu sifat dari bahas Golang adalah perlu parameter yang match 100% antara pemanggil fungsi/prosedur dengan pemanggilnya.