

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

Modul 15 Test



Disusun Oleh :

Egi Umar Ferdhika / 2311102277

11-IF-06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. Soal Studi Case

- 1) Sebuah program yang digunakan untuk mencari sebuah irisan himpunan.

Masukan terdiri dari dua baris. Setiap barisnya yang berisi sekumpulan bilangan. Masukan disetiap barisnya akan berakhir apabila bilangan yang diberikan sudah pernah diberikan pada baris tersebut (atau duplikat). Catatan: anggota suatu himpunan tidak boleh duplikat.

Keluaran adalah sekumpulan bilangan yang menyatakan irisan dari himpunan pada baris pertama dan baris kedua pada masukan.

Contoh masukan dan keluaran:

No.	Masukan	Keluaran	Penjelasan
1	11 28 33 64 95 16 100 15 64 3 11 7 28 33 6 28	11 28 33	64 dan 28 duplikat sehingga masukan berhenti dan tidak perlu dimasukkan ke dalam array
2	1 1 1 1	1	
3	1 2 3 4 3 9 8 7 9		

```
package main
import "fmt"

type set [2022]int

func exist(T set, n int, val int) bool
/* mengembalikan true apabila bilangan val ada di dalam array T yang berisi
sejumlah n bilangan bulat */

func inputSet(T *set, n *int)
/* I.S. data himpunan telah siap pada piranti masukan
F.S. array T berisi sejumlah n bilangan bulat yang berasal dari masukan
(masukan berakhir apabila bilangan ada yang duplikat, atau array penuh)
Catatan: Panggil function exist di sini untuk membantu pengecekan */

func findIntersection(T1,T2 set, n,m int, T3 *set, h *int)
/* I.S. terdefinisi himpunan T1 dan T2 yang berisi sejumlah n dan m anggota
himpunan
F.S. himpunan T3 berisi sejumlah h bilangan bulat yang merupakan irisan
dari himpunan T1 dan T2
Catatan: Panggil function exist di sini untuk membantu pengecekan */

func printSet(T set, n int)
/* I.S. terdefinisi sebuah himpunan T yang berisi sejumlah n bilangan bulat
```

Halaman 92 | Modul Praktikum Algoritma dan Pemrograman 2

```
F.S. menampilkan isi array T secara horizontal (dipisahkan oleh spasi) */

func main(){
    var s1,s2,s3 set
    var n1,n2,n3 int
    inputSet(&s1,&n1)
    inputSet(&s2,&n2)
    findIntersection(s1,s2,n1,n2,&s3,&n3)
    printSet(s3,n3)
}
```

Sourcecode

```

package main

import (
    "fmt"
)

const maxSize = 2022

type set [maxSize]int

Tabnine | Edit | Test | Explain | Document | Ask
func exist(T set, n int, val int) bool {
    for i := 0; i < n; i++ {
        if T[i] == val {
            return true
        }
    }
    return false
}

Tabnine | Edit | Test | Explain | Document | Ask
func inputSet(T *set, n *int) {
    fmt.Println("Masukkan bilangan untuk himpunan (akhir dengan duplikat):")
    var val int
    *n = 0

    for {
        _, err := fmt.Scan(&val)
        if err != nil {
            break
        }

        if exist(*T, *n, val) {
            break
        }

        if *n < maxSize {
            T[*n] = val
            *n++
        } else {
            break
        }
    }
}

Tabnine | Edit | Test | Explain | Document | Ask
func findIntersection(T1, T2 set, n1, n2 int, T3 *set, n3 *int) {
    *n3 = 0
    for i := 0; i < n1; i++ {
        if exist(T2, n2, T1[i]) {
            if *n3 < maxSize {
                T3[*n3] = T1[i]
                *n3++
            }
        }
    }
}

```

```

func printSet(T set, n int) {
    for i := 0; i < n; i++ {
        if i > 0 {
            fmt.Print(" ")
        }
        fmt.Print(T[i])
    }
    fmt.Println()
}

Tabnine | Edit | Test | Explain | Document | Ask
func main() {
    var s1, s2, s3 set
    var n1, n2, n3 int

    inputSet(&s1, &n1)
    inputSet(&s2, &n2)
    findIntersection(s1, s2, n1, n2, &s3, &n3)
    fmt.Println("Irisan himpunan:")
    printSet(s3, n3)
}

```

OUTPUT:

```

Masukkan bilangan untuk himpunan (akhir dengan duplikat):
11 28 33 64 95 16 100 15 64
Masukkan bilangan untuk himpunan (akhir dengan duplikat):
3 11 7 28 33 6 28
Irisan himpunan:
11 28 33

```

II. Soal Studi Case

- 2) Suatu tabel digunakan untuk mencatat data mahasiswa. Mahasiswa memiliki atribut NIM, nama, dan nilai. Setiap data baru akan selalu ditambahkan ke dalam tabel di indeks N+1. N adalah jumlah data saat ini di dalam array. Sehingga pada tabel mungkin terdapat beberapa data untuk seorang mahasiswa. Contoh isi tabel sebagai berikut:

114, Nana, 97	113, Jojo, 70	118, Rere, 88	116, Koko, 40	117, Keke, 90	116, Koko, 60	113, Jojo, 50	113, Jojo, 80	118, Rere, 88	119, Roro, 100
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	----------------------

Pada contoh di atas, data Jojo ada tiga dengan 70 sebagai nilai pertama, kemudian 50, dan 80 sebagai nilai terakhir.

Definisikan struct dan array berikut:

```
constant nMax: integer = 51
type mahasiswa <NIM: string, nama:string, nilai:integer>
type arrayMahasiswa: array [1..nMax] of mahasiswa
```

Kemudian buatlah program dengan spesifikasi berikut:

- Menerima masukan sejumlah N data mahasiswa kemudian menyimpannya ke dalam array. N diperoleh dari masukan.
- Fungsi untuk mencari nilai pertama seorang mahasiswa dengan NIM tertentu.
- Fungsi untuk mencari nilai terbesar seorang mahasiswa dengan NIM tertentu.
- Menampilkan hasil pencarian pada poin a dan b.

Sourcecode

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 const nMax = 51
8
9 type Mahasiswa struct {
10     NIM    string
11     Nama   string
12     Nilai  int
13 }
14 type ArrayMahasiswa [nMax]Mahasiswa
15
16 Tabnine | Edit | Test | Explain | Document | Ask
17 func cariNilaiPertama(data ArrayMahasiswa, nim string) (int, bool) {
18     for _, m := range data {
19         if m.NIM == nim {
20             return m.Nilai, true
21         }
22     }
23     return 0, false
24 }
25
26 Tabnine | Edit | Test | Explain | Document | Ask
27 func cariNilaiTerbesar(data ArrayMahasiswa, nim string) (int, bool) {
28     terbesar := -1
29     found := false
30     for _, m := range data {
31         if m.NIM == nim {
32             if m.Nilai > terbesar {
33                 terbesar = m.Nilai
34                 found = true
35             }
36         }
37     }
38     return terbesar, found
39 }
40
41 Tabnine | Edit | Test | Explain | Document | Ask
42 func main() {
43     var dataMahasiswa ArrayMahasiswa
44     var N int
45     fmt.Print("Masukkan jumlah mahasiswa (maksimal 51): ")
46     fmt.Scan(&N)
47
48     if N < 1 || N > nMax {
49         fmt.Println("Jumlah mahasiswa harus antara 1 dan 51.")
50         return
51     }
52 }

```

```

49     for i := 0; i < N; i++ {
50         fmt.Printf("Masukkan NIM mahasiswa ke-%d: ", i+1)
51         fmt.Scan(&dataMahasiswa[i].NIM)
52         fmt.Printf("Masukkan nama mahasiswa ke-%d: ", i+1)
53         fmt.Scan(&dataMahasiswa[i].Nama)
54         fmt.Printf("Masukkan nilai mahasiswa ke-%d: ", i+1)
55         fmt.Scan(&dataMahasiswa[i].Nilai)
56     }
57     var nimCari string
58     fmt.Print("Masukkan NIM yang ingin dicari: ")
59     fmt.Scan(&nimCari)
60
61     nilaiPertama, ditemukanPertama := cariNilaiPertama(dataMahasiswa, nimCari)
62     if ditemukanPertama {
63         fmt.Printf("Nilai pertama mahasiswa dengan NIM %s adalah: %d\n", nimCari, nilaiPertama)
64     } else {
65         fmt.Printf("Mahasiswa dengan NIM %s tidak ditemukan.\n", nimCari)
66     }
67     nilaiTerbesar, ditemukanTerbesar := cariNilaiTerbesar(dataMahasiswa, nimCari)
68     if ditemukanTerbesar {
69         fmt.Printf("Nilai terbesar mahasiswa dengan NIM %s adalah: %d\n", nimCari, nilaiTerbesar)
70     } else {
71         fmt.Printf("Mahasiswa dengan NIM %s tidak ditemukan.\n", nimCari)
72     }
73 }
74

```

OUTPUT:

```

Masukkan jumlah mahasiswa (maksimal 51): 3
Masukkan NIM mahasiswa ke-1: 1
Masukkan nama mahasiswa ke-1: herman
Masukkan nilai mahasiswa ke-1: 98
Masukkan NIM mahasiswa ke-2: 2
Masukkan nama mahasiswa ke-2: sugeng
Masukkan nilai mahasiswa ke-2: 90
Masukkan NIM mahasiswa ke-3: 3
Masukkan nama mahasiswa ke-3: malek
Masukkan nilai mahasiswa ke-3: 87
Masukkan NIM yang ingin dicari: 2
Nilai pertama mahasiswa dengan NIM 2 adalah: 90
Nilai terbesar mahasiswa dengan NIM 2 adalah: 90

```

III. Soal Studi Case

- 3) Sebuah program digunakan untuk mengolah data nama provinsi, populasi, dan angka pertumbuhan penduduk provinsi di Indonesia pada tahun 2018.

Halaman 93 | Modul Praktikum Algoritma dan Pemrograman 2

Masukan terdiri dari 35 baris, yang mana masing-masing barisnya terdiri dari tiga nilai yang menyatakan nama provinsi, jumlah populasi provinsi (bilangan bulat), dan angka pertumbuhan (riil) provinsi tersebut. Pada baris terakhir hanya sebuah string yang menyatakan nama provinsi yang akan dicari.

Keluaran terdiri dari 36 baris. Baris pertama adalah nama provinsi dengan angka pertumbuhan tercepat. Baris kedua adalah indeks provinsi yang dicari sesuai dengan nama provinsi yang ditulis pada masukan baris terakhir. Terakhir terdiri dari 34 baris yang menampilkan nama provinsi beserta prediksi jumlah penduduk pada provinsi tersebut di tahun depannya, khusus yang memiliki pertumbuhan di atas 2%.

Lengkapi program berikut sesuai dengan spesifikasi dari subprogram yang diberikan.

```
program Provinsi
kamus
    const nProv : integer = 34
    type NamaProv = array [1..nProv] of string
    type PopProv = array [1..nProv] of integer
    type TumbuhProv = array [1..nProv] of real
algoritma
    ...
endprogram

procedure InputData(in/out prov:NamaProv, pop:PopProv, tumbuh:TumbuhProv )
(I.S. Data-data provinsi tersedia pada input device
 F.S. Array prov, pop, dan tumbuh berisi data yang diberikan.)

function ProvinsiTercepat( tumbuh : TumbuhProv ) → integer
(Mengembalikan indeks array tumbuh dengan pertumbuhan penduduk tercepat)

procedure Prediksi (in prov:NamaProv, in pop:PopProv, in tumbuh:TumbuhProv)
(I.S. Tabel prov, pop, dan tumbuh berisi data-data provinsi
 F.S. Tampilan seluruh nama provinsi dan prediksi jumlah penduduknya di tahun
depan dengan pertumbuhan diatas 0.02 (atau diatas 2%) )

function IndeksProvinsi(prov:NamaProvinsi, nama:string) → integer
(Mengembalikan indeks array prov untuk provinsi dengan nama tersebut, dan -1
jika tidak ada provinsi dengan nama tersebut )
```

Sourcecode

Modul 15 > 3 > 3.go > ...

```
1  package main
2
3  import (
4      "fmt"
5      "math"
6  )
7
8  const nProv = 35
9
10 type Provinsi struct {
11     nama      string
12     populasi  int
13     tumbuh    float64
14 }
15
16 var data [nProv]Provinsi
17 var namaDicari string
18
19 Tabnine | Edit | Test | Explain | Document | Ask
20 func InputData() {
21     for i := 0; i < nProv; i++ {
22         fmt.Scan(&data[i].nama, &data[i].populasi, &data[i].tumbuh)
23     }
24     fmt.Scan(&namaDicari)
25 }
26
27 Tabnine | Edit | Test | Explain | Document | Ask
28 func ProvinsiTercepat() int {
29     maxTumbuh := data[0].tumbuh
30     index := 0
31     for i := 1; i < nProv; i++ {
32         if data[i].tumbuh > maxTumbuh {
33             maxTumbuh = data[i].tumbuh
34             index = i
35         }
36     }
37     return index
38 }
```

```

Tabnine | Edit | Test | Explain | Document | Ask
38 func Prediksi() {
39     for i := 0; i < nProv; i++ {
40         if data[i].tumbuh > 0.02 {
41             prediksi := int(math.Round(float64(data[i].populasi) * (1 + data[i].tumbuh)))
42             fmt.Printf("%s %d\n", data[i].nama, prediksi)
43         }
44     }
45 }

Tabnine | Edit | Test | Explain | Document | Ask
47 func IndeksProvinsi(nama string) int {
48     for i := 0; i < nProv; i++ {
49         if data[i].nama == nama {
50             return i
51         }
52     }
53     return -1
54 }

Tabnine | Edit | Test | Explain | Document | Ask
56 func main() {
57     InputData()
58
59     indeksTercepat := ProvinsiTercepat()
60     fmt.Println(data[indeksTercepat].nama)
61
62     indeksDicari := IndeksProvinsi(namaDicari)
63     fmt.Println(indeksDicari)
64
65     Prediksi()
66 }

```

OUTPUT:

```

alpro 2\eval 1\Modul 15\3\3.go
3\3.go:1:1: expected 'package', found 'EOF'
PS D:\Kuliah\SEM 3\parktikum alpro 2\eval 1\Modul 15>

```

IV. Soal Studi Case

- 4) Kompetisi pemrograman yang baru saja berlalu diikuti oleh 17 tim dari berbagai perguruan tinggi ternama. Dalam kompetisi tersebut, setiap tim berlomba untuk menyelesaikan sebanyak mungkin problem yang diberikan. Dari 13 problem yang diberikan, ada satu problem yang menarik. Problem tersebut mudah dipahami, hampir semua tim mencoba untuk menyelesaikannya, tetapi hanya 3 tim yang berhasil. Apa sih problemnya?

Halaman 94 | Modul Praktikum Algoritma dan Pemrograman 2

"Median adalah nilai tengah dari suatu koleksi data yang sudah terurut. Jika jumlah data ganap, maka nilai median adalah rerata dari kedua nilai tengahnya. Pada problem ini, semua data merupakan bilangan bulat positif, dan karenanya rerata nilai tengah dibulatkan kebawah."

Buatlah program median yang mencetak nilai median terhadap seluruh data yang sudah terbaca, jika data yang dibaca saat itu adalah 0.

Masukan berbentuk rangkaian bilangan bulat. Masukan tidak akan berisi lebih dari 1000000 data, tidak termasuk bilangan 0. Data 0 merupakan tanda bahwa median harus dicetak, tidak termasuk data yang dicari mediannya. Data masukan diakhiri dengan bilangan bulat -5313541.

Keluaran adalah median yang diminta, satu data perbaris.

Petunjuk:

- Untuk setiap data bukan 0 (dan bukan marker -5313541) simpan ke dalam array.
- dan setiap kali menemukan bilangan 0, urutkanlah data yang sudah tersimpan dengan menggunakan metoda selection sort dan ambil mediannya.

Contoh masukan dan keluaran

1	Masukan	7 23 11 0 5 19 2 29 3 13 17 0 -5313541
	Keluaran	11 12
	Penjelasan	Sampai bilangan 0 yang pertama, data terbaca adalah 7 23 11, setelah tersusun: 7 11 23, maka median saat itu adalah 11. Sampai bilangan 0 yang kedua, data adalah 7 23 11 5 19 2 29 3 13 17, setelah tersusun diperoleh: 2 3 5 7 11 13 17 19 23 29. Karena ada 10 data, ganap, maka median adalah $(11+13)/2=12$.
2	Masukan	23 12 24 26 28 18 10 25 8 0 -5313541
	Keluaran	21.5
	Penjelasan	Sampai bilangan 0 yang pertama, data terbaca adalah 23 12 24 26 28 18 25 8, setelah tersusun: 8 10 12 20 23 24 25 26, maka median saat itu adalah $(20+23)/2=21.5$.

```
1 package main
2 import "fmt"
3
4 const NMAX = ...
5 type arrInt [...]int
6
7 func sorting(T *arrInt, n int){
8     /* I.S. terdefinisi array T yang berisi sejumlah n bilangan bulat
9      * F.S. array T terurut secara membesar berdasarkan algoritma selection
10    sort */
11    ...
12 }
13
14 func median(T arrInt, n int) float64 {
15
```

Sourcecode

```

1  package main
2
3  import (
4      "fmt"
5      "sort"
6  )
7
8  const NMAX = 1000000
9
10 type arrInt []int
11
12 Tabnine | Edit | Test | Explain | Document | Ask
13 func main() {
14     var nomor int
15     var nomorS []int
16     fmt.Println("Masukkan bilangan (akhiri dengan -5313541):")
17     for {
18         fmt.Scan(&nomor)
19
20         if nomor == -5313541 {
21             break
22         }
23
24         if nomor == 0 {
25             if len(nomorS) > 0 {
26                 sort.Ints(nomorS)
27                 median := median(nomorS)
28                 fmt.Printf("Median: %d\n", median)
29             }
30         } else {
31             nomorS = append(nomorS, nomor)
32         }
33     }
34 }
35
36 Tabnine | Edit | Test | Explain | Document | Ask
37 func median(nomorS []int) int {
38     length := len(nomorS)
39     if length%2 == 1 {
40         return nomorS[length/2]
41     }
42     return (nomorS[(length/2)-1] + nomorS[length/2]) / 2
43 }

```

OUTPUT:

```

Masukkan bilangan (akhiri dengan -5313541):
7 23 11 0 5 19 2 29 3 13 17 0 -5313541
Median: 11
Median: 12

```

V. Soal Studi Case

- 4) Kompetisi pemrograman yang baru saja berlalu diikuti oleh 17 tim dari berbagai perguruan tinggi ternama. Dalam kompetisi tersebut, setiap tim berlomba untuk menyelesaikan sebanyak mungkin problem yang diberikan. Dari 13 problem yang diberikan, ada satu problem yang menarik. Problem tersebut mudah dipahami, hampir semua tim mencoba untuk menyelesaikannya, tetapi hanya 3 tim yang berhasil. Apa sih problemnya?

halaman 94 | Modul Praktikum Algoritma dan Pemrograman 2

"Median adalah nilai tengah dari suatu koleksi data yang sudah terurut. Jika jumlah data genap, maka nilai median adalah rerata dari kedua nilai tengahnya. Pada problem ini, semua data merupakan bilangan bulat positif, dan karenanya rerata nilai tengah dibulatkan kebawah."

Buatlah program median yang mencetak nilai median terhadap seluruh data yang sudah terbaca, jika data yang dibaca saat itu adalah 0.

Masukan berbentuk rangkaian bilangan bulat. Masukan tidak akan berisi lebih dari 1000000 data, tidak termasuk bilangan 0. Data 0 merupakan tanda bahwa median harus dicetak, tidak termasuk data yang dicari mediannya. Data masukan diakhiri dengan bilangan bulat -5313541.

Keluaran adalah median yang diminta, satu data perbaris.

Petunjuk:

- Untuk setiap data bukan 0 (dan bukan marker -5313541) simpan ke dalam array.
- dan setiap kali menemukan bilangan 0, urutkanlah data yang sudah tersimpan dengan menggunakan metoda selection sort dan ambil mediannya.

Contoh masukan dan keluaran

1	Masukan	7 23 11 0 5 19 2 29 3 13 17 0 -5313541
	Keluaran	11 12
	Penjelasan	Sampai bilangan 0 yang pertama, data terbaca adalah 7 23 11, setelah tersusun: 7 11 23, maka median saat itu adalah 11. Sampai bilangan 0 yang kedua, data adalah 7 23 11 5 19 2 29 3 13 17, setelah tersusun diperoleh: 2 3 5 7 11 13 17 19 23 29. Karena ada 10 data, genap, maka median adalah $(11+13)/2=12$.
2	Masukan	23 12 24 26 28 10 25 0 -5313541
	Keluaran	21.5
	Penjelasan	Sampai bilangan 0 yang pertama, data terbaca adalah 23 12 24 26 28 10 25 0, setelah tersusun: 8 10 12 20 23 24 25 26, maka median saat itu adalah $(20+23)/2=21.5$.

```
1 package main
2 import "fmt"
3
4 const NMAX = ...
5 type arrInt [...]int
6
7 func sorting(T *arrInt, n int){
8     /* I.S. terdefinisi array T yang berisi sejumlah n bilangan bulat
9      * F.S. array T terurut secara membesar berdasarkan algoritma selection
10     sort */
11     ...
12 }
13
14 func median(T arrInt, n int) float64 {
15
```

Sourcecode

```

package main

import (
    "fmt"
    "sort"
)

const NMAX = 1000000

type arrInt []int

Tabnine | Edit | Test | Explain | Document | Ask
func main() {
    var suara int
    var partai []int

    fmt.Println("Masukkan bilangan (akhiri dengan -1):")
    for {
        fmt.Scan(&suara)

        if suara == -1 {
            break
        }

        if suara == 0 {
            if len(partai) > 0 {
                sort.Ints(partai)
                posisi := posisi(partai)
                fmt.Printf("posisi: %d\n", posisi)
            }
        } else {
            partai = append(partai, suara)
        }
    }
}

Tabnine | Edit | Test | Explain | Document | Ask
func posisi(partai []int) int {
    length := len(partai)
    if length%2 == 1 {
        return partai[length/2]
    }
    return (partai[(length/2)-1] + partai[length/2]) / 2
}

```

OUTPUT:

```
Masukkan bilangan (akhiri dengan -1):  
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 -1  
PS D:\Kuliah\SEM 3\parktikum alpro 2\eval 1\Modul 15>
```