

# CII1F4/CPI1F4 PRAKTIKUM 12 2020/2021

## JURNAL PRAKTIKUM - SORTING

**Petunjuk praktikum:** Buatlah program Go sesuai dengan materi yang telah dipelajari dan diajarkan. Asisten memberikan tutorial apabila ada praktikan yang mengalami kesulitan.

### SOAL

1. **[tie-break]** Di masa pandemi covid-19 ini English Premier League sudah memasuki minggu ke-11. Tanpa kehadiran langsung fans fanatik di stadion, semua klub terasa berat melakoni pertandingan demi pertandingan. Klub tidak lagi memperoleh pemasukan dari hasil penjualan tiket. Selain itu tanpa dukungan langsung penonton, gairah pemain menunjukkan kemampuan mereka secara optimal jauh berkurang. Tidak ada gol-gol spektakular atau permainan-permainan menggocek bola yang cantik.

Untuk sedikit mendorong gairah pemain, penyelenggara liga ingin menampilkan peringkat pencetak gol terbanyak sementara. Data itu ingin disusun secara menurun terhadap perolehan gol dan assist. Atribut data adalah nama, jumlah gol, dan jumlah assist. Jika ada dua orang pemain atau lebih memiliki jumlah gol yang sama, maka assist yang lebih banyak akan meletakkan pemain tersebut pada peringkat lebih atas.

**Masukan** terdiri atas **n** (tipe data integer) yang menyatakan jumlah pencetak gol, diikuti sebanyak **n** baris berikutnya berupa nama (tipe data string), jumlah gol (tipe data integer), dan jumlah assist (tipe data integer).

**Keluaran** terdiri atas data pemain (nama, jumlah goal, jumlah assist) yang tersusun menurun sesuai dengan kriteria diatas.

Untuk kebutuhan pembuatan program, algoritma sorting yang boleh digunakan adalah Selection sort atau Insertion sort. Selain itu anggap bahwa jumlah pemain yang diperhitungkan dalam peringkat tidak lebih dari 1001 orang pemain saja.

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	9 Bruno Fernandes 7 3 Jamie Vardy 8 1 Dominic Calvert-Lewin 10 2 Son Heung-Min 9 2 Harry Kane 7 2	Dominic Calvert-Lewin 10 2 Son Heung-Min 9 2 Mohamed Salah 8 2 Jamie Vardy 8 1 Bruno Fernandes 7 3 Harry Kane 7 2

Ollie Watkins 6 1	Patrick Bamford 7 1
Patrick Bamford 7 1	Callum Wilson 7 0
Callum Wilson 7 0	Ollie Watkins 6 1
Mohamed Salah 8 2	

2. [PilLeg] Sebuah program digunakan untuk menghitung perolehan suara dari berbagai partai politik dalam sebuah pemilihan umum calon legislatif. Program akan menampilkan data partai terurut berdasarkan perolehan suara terurut. Nama partai hanya disimbolkan dari angka 1 hingga N, di mana  $1 \leq N \leq 1000000$ .

**Masukan** berupa sejumlah nilai yang dipisahkan spasi. Masing-masing nilai menyatakan nama partai (1 hingga N) yang dipilih. Proses input ini diakhiri dengan nilai -1.

**Keluaran** berupa daftar partai dan peroleh suaranya yang terurut descending atau mengecil dengan format <partai>(<suara>). Perhatikan contoh masukan dan keluaran yang diberikan.

**Petunjuk** : gunakan tipe bentukan partai yang berisi nama dan suara. Data perolehan suara disimpan pada array of partai (kapasitas 1000000). Array tersebutlah yang diurutkan.

**Lengkapi program berikut ini**

```
package main
import "fmt"
const NMAX = 1000000
// tipe bentukan partai
...
// tipe array of partai dengan kapasitas NMAX
...

func main(){
// deklarasi variabel
...
// lakukan proses input suara secara berulang di sini, simpan ke dalam
array p, sehingga terdapat array p yang berisi hasil peroleh suara n
partai.
...
// lakukan proses pengurutan dengan insertion sort descending berdasarkan
jumlah suara yang diperoleh
...

// tampilkan array p
...
}

func posisi(t tabPartai, n int, nama int) int {
```

```
/* mengembalikan indeks partai yang memiliki nama yang dicari pada array t
yang berisi n partai atau -1 apabila tidak ditemukan , gunakan pencarian
sekuensial */
```

#### Contoh masukan dan keluaran

No	Masukan	Keluaran
1	5 1 1 1 1 1 1 1 3 3 3 3 3 2 2 5 5 5 5 5 4 3 2 2 2 2 -1	1(7) 5(6) 3(6) 2(6) 4(1)
2	5 8 8 5 6 8 8 7 6 5 8 7 5 6 7 5 8 6 7 8 8 7 7 8 6 7 7 6 8 6 8 8 5 5 6 6 6 7 7 6 7 8 8 8 5 7 6 6 8 6 5 5 8 7 5 5 6 8 7 6 5 5 8 6 6 7 8 8 8 6 7 6 6 5 7 8 7 6 6 6 8 7 7 8 6 5 5 7 7 6 5 7 8 8 6 8 8 6 7 8 -1	8(30) 6(28) 7(24) 5(18)
3	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 -1	8(15)
4	10 1 7 8 10 1 4 8 8 5 -1	8(3) 10(2) 1(2) 7(1) 4(1) 5(1)
5	14 10 13 13 14 10 11 13 13 12 15 11 10 -1	13(4) 10(3) 14(2) 11(2) 12(1) 15(1)
6	-1	

3. [Tanggal] Buatlah sebuah program sederhana digunakan untuk mengurutkan tanggal dengan format <year>.<month>.<day>. Di mana <year> adalah 4 digit tahun, <month> adalah 2 digit bulan, dan <day> adalah 2 digit tanggal.

**Masukan** berupa beberapa baris string yang menyatakan tanggal (dengan format sesuai penjelasan di atas). Proses input berhenti apabila tanggal #####.##.##

**Keluaran** berupa beberapa baris string tanggal yang telah terurut secara mengecil atau descending.

**Lengkapi potongan program berikut sehingga membentuk program lengkap!**

```
Program tanggal
kamus
    constant NMAX : integer = 1000000
    type tabDate = array [0..NMAX-1] of string
    t : tabDate
    n : integer
algoritma
    {panggil procedure addData di sini}
    ...
    {panggil procedure urut di sini}
    ...
    {panggil procedure printData di sini}
    ...
endprogram

procedure addData(in/out list : tabDate, in/out n : integer)
{ IS. sejumlah n data tanggal telah siap pada piranti masukan
  FS. array list berisi n data tanggal }
```

```

procedure urut(in/out list : tabDate, in n : integer)
{ IS. terdefinisi sejumlah n tanggal pada array list
  FS. array list terurut secara descending atau mengecil menggunakan
  selection sort. Catatan: gunakan subprogram maxPos dan swap pada procedure
  ini}

function maxPos(list:tabDate, start,n:integer) -> integer
{ IS. list berisi n data tanggal, dan start yang berisi index awal pencarian
  nilai maksimum
  FS. mengembalikan indeks nilai maksimum yang terdapat pada array list dari
  indeks ke-start sampai ke n-1}

procedure swap(in/out x,y : string){
{ IS. terdefinisi dua buah string x dan y
  FS. nilai x dan y saling tertukar sama lainnya }

procedure printData(list: tabDate, n: integer){
{ IS. terdefinisi sejumlah n tanggal pada array list
  FS. menampilkan array list ke layar }

```

#### Contoh masukan dan keluaran

No	Masukan	Keluaran
1	2008.01.14 2005.10.01 1996.08.27 2005.05.21 2015.03.02 2006.08.11 1992.04.24 2001.01.14 2017.03.20 2001.07.14 ####.##.##	2017.03.20 2015.03.02 2008.01.14 2006.08.11 2005.10.01 2005.05.21 2001.07.14 2001.01.14 1996.08.27 1992.04.24
2	2020.10.16 2020.10.15 2020.10.21 2020.10.17 2020.10.03 2020.10.02 2020.10.20 2020.10.14 2020.10.15 2020.10.22 ####.##.##	2020.10.22 2020.10.21 2020.10.20 2020.10.17 2020.10.16 2020.10.15 2020.10.15 2020.10.14 2020.10.03 2020.10.02
3	####.##.##	
4	2020.07.05 2020.04.05 2020.05.05 2020.09.05 2020.02.05 2020.01.05 2020.10.05 2020.07.05 2020.08.05 2020.11.05	2020.11.05 2020.10.05 2020.10.05 2020.10.05 2020.09.05 2020.09.05 2020.09.05 2020.08.05 2020.08.05 2020.08.05

	2020.08.05	2020.07.05
	2020.10.05	2020.07.05
	2020.10.05	2020.07.05
	2020.03.05	2020.06.05
	2020.09.05	2020.05.05
	2020.06.05	2020.04.05
	2020.08.05	2020.03.05
	2020.09.05	2020.02.05
	2020.07.05	2020.01.05
	2020.01.05	2020.01.05
	####.##.##	