

## Modul O2 Terbimbing

### Fungsi

1. **[Vokal]** Buatlah program dalam bahasa Go untuk menghitung banyaknya vokal dalam suatu rangkaian karakter berjenis alfabet ['a'..'z'] dan ['A'..'Z']. Program yang dibuat memanfaatkan fungsi bernama **vowel** yang mengecek apakah suatu karakter merupakan vokal atau bukan.

```
function vowel(kar : char) -> boolean
{mengembalikan true apabila kar adalah vokal, dan false apabila
sebaliknya}
```

Masukan dan keluaran program adalah sebagai berikut:

**Masukan** terdiri dari sekumpulan karakter alfabet yang diakhiri oleh karakter titik ' . '

**Keluaran** berupa teks yang menyatakan banyaknya vokal.

Petunjuk: Gunakan tipe byte untuk deklarasi char.

#### Contoh masukan dan keluaran

No	Masukan	Keluaran
1	POKEMON.	Jumlah vokal adalah 3
2	EkoNOmi.	Jumlah vokal adalah 4
3	INforMATika.	Jumlah vokal adalah 5
4	Hdd.	Jumlah vokal adalah 0

2. **[Diskon dan cashback Konoha]** Di Desa Konoha, sebuah toko peralatan ninja mengadakan promo besar-besaran untuk menyambut pembukaan cabang baru di desa tersebut. Setiap pembelian senjata atau perlengkapan ninja di toko tersebut akan diberikan poin yang dapat ditukar sebagai cashback dan diskon. Buatlah program untuk menghitung persentase diskon dan cashback yang diperoleh pembeli dengan masukan dan keluaran sebagai berikut:

**Masukan** terdiri dari status keanggotaan tim (string "yes" atau "no") dan lima bilangan bulat a, b, c, d, dan e yang menyatakan poin yang diperoleh.

**Keluaran** terdiri dari dua bilangan real yang menyatakan persentase diskon dan cashback yang diperoleh.

Program akan memanfaatkan empat buah fungsi. Aturan perhitungan diskon dan cashback terdapat pada spesifikasi fungsi yang diberikan.

```
function ganjil(a, b, c, d, e : integer) -> boolean  
{ Mengembalikan true apabila masing-masing a, b, c, d, dan e adalah ganjil }
```

```
function genap(a, b, c, d, e : integer) -> boolean  
{ Mengembalikan true apabila masing-masing a, b, c, d, dan e adalah genap }
```

```
function diskon(member : string, c, d, e : integer) -> real  
{ Mengembalikan nilai diskon sebesar 1.7x dari total c, d, dan e. Tambahkan 15% dari nilai diskon awal apabila member bernilai "yes", nilai diskon maksimal adalah 50% }
```

```
function cashback(member : string, a, b, c : integer) -> real  
{ Mengembalikan nilai cashback sebesar 3.1x dari total a, b, dan c dengan a, b, dan c adalah poin. Tambahkan 15% dari nilai cashback awal apabila member bernilai "yes", nilai cashback maksimal adalah 35% }
```

#### **program** Promo

{ Program ini akan menampilkan persentase diskon dan cashback yang diperoleh, apabila diberikan status membership m dan lima poin yang diperoleh a, b, c, d, dan e. Aturan pemberian diskon dan cashback:

- Diskon diberikan apabila lima poin bernilai ganjil semua, dan besarnya diskon dihitung dari tiga poin terakhir (c, d, dan e).
- Cashback diberikan apabila lima poin bernilai genap semua, dan besarnya cashback dihitung dari tiga poin pertama (a, b, dan c)

- Diskon dan cashback diberikan apabila lima poin ada yang bernilai ganjil dan genap.}

Persentase diskon dan cashback dicetak dengan pembulatan 2 digit di belakang koma

#### Contoh masukan dan keluaran

No	Masukan	Keluaran
1	yes 1 2 3 4 5	23.46 21.39
2	no 1 2 3 4 5	20.40 18.60
3	yes 3 3 3 3 3	17.59 0.00
4	no 6 4 2 4 2	0.00 35.00

## Modul O2 Mandiri

### Fungsi

1. **[Persamaan matematika]** Apabila didefinisikan  $f(x, y, z) = ((2x)/(x + y + z))$  dan  $g(x, y) = ((1/(5x)) + y)$ , maka buatlah subprogram fungsi berikut berdasarkan spesifikasi yang diberikan. Catatan: Perhatikan urutan parameter seperti terlihat pada contoh.

```
function fungsi_f(x, y, z : real) -> real
```

```
{ Fungsi menerima masukan x, y, dan z serta mengembalikan nilai real berdasarkan persamaan  $f(x, y, z) = ((2x)/(x + y + z))$  }
```

```
function fungsi_g(x, y : real) -> real
```

```
{ Fungsi menerima masukan x, dan y, serta mengembalikan nilai real berdasarkan persamaan  $g(x, y) = ((1/(5x)) + y)$  }
```

Selanjutnya kedua fungsi dipanggil dalam program dengan masukan dan keluaran sebagai berikut:

**Masukan** berupa tiga bilangan real b1, b2, dan b3. Ketiganya akan menjadi parameter aktual terhadap parameter formal x, y, dan z.

**Keluaran** berupa dua bilangan real hasil perhitungan fungsi f dan fungsi g.

#### Contoh masukan dan keluaran

No	Masukan	Keluaran
1	1.5 2.4 3.1	0.42857142857142855 2.4
2	100 45 50	1.0256410256410255 45

2. **[Kunai Kakashi]** Kakashi Hatake sedang berlatih melemparkan kunai. Kakashi melemparkan kunai dengan sudut elevasi **T** derajat dan kecepatan awal **V** m/s. Buatlah program dalam bahasa Go untuk menghitung:

- Waktu yang diperlukan kunai untuk mencapai titik tertinggi. Gunakan rumus  $t = V \sin(T) / g$ , dengan  $t$  adalah waktu,  $V$  adalah kecepatan,  $T$  adalah sudut elevasi, dan  $g$  adalah konstanta gravitasi.
- Jarak horizontal yang ditempuh kunai sebelum mencapai tanah. Gunakan rumus  $x = V^2 \sin(2T) / g$ , dengan  $x$  adalah jarak,  $V$  adalah kecepatan,  $T$  adalah sudut elevasi, dan  $g$  adalah konstanta gravitasi.
- Ketinggian maksimum yang dicapai oleh kunai. Gunakan rumus  $y = V^2 \sin^2(T) / 2g$ , dengan  $y$  adalah ketinggian,  $V$  adalah kecepatan,  $T$  adalah sudut elevasi, dan  $g$  adalah konstanta gravitasi.

Catatan: Konstanta gravitasi  $g$  adalah  $9.8 \text{ m/s}^2$ .

Program dibuat dengan memanfaatkan fungsi-fungsi berikut:

**function** konversiDerajatkeRadian( $T : \text{real}$ ) -> **real**

{ Mengembalikan nilai konversi sudut dari  $T$  derajat ke radian. Sudut satu derajat setara dengan satu  $\pi/180$  radian. Gunakan nilai  $\pi$  dari `math.Pi` }

**function** waktu( $V, T : \text{real}$ ) -> **real**

{ Mengembalikan nilai waktu yang diperlukan untuk mencapai posisi tertinggi, jika kunai dilempar dengan sudut  $T$  derajat dan kecepatan awal  $V \text{ m/s}$ . Gunakan fungsi **konversiDerajatkeRadian** untuk mengubah sudut dari derajat ke radian. }

**function** jarak( $V, T : \text{real}$ ) -> **real**

{ Mengembalikan nilai jarak horizontal yang ditempuh kunai sebelum mencapai tanah, jika kunai dilempar dengan sudut  $T$  derajat dan kecepatan awal  $V \text{ m/s}$ . Gunakan fungsi **konversiDerajatkeRadian** untuk mengubah sudut dari derajat ke radian. }

**function** ketinggian(V, T : real) -> real

{ Mengembalikan nilai ketinggian maksimum yang dicapai kunai, jika kunai dilempar dengan sudut T derajat dan kecepatan awal V m/s. Gunakan fungsi **konversiDerajatkeRadian** untuk mengubah sudut dari derajat ke radian. }

Masukan dan keluaran program adalah sebagai berikut:

**Masukan** berupa dua bilangan real V dan T yang menyatakan kecepatan awal lemparan dan sudut elevasi dalam derajat.

**Keluaran** berupa tiga bilangan real yang menyatakan waktu tempuh, jarak horizontal, dan ketinggian maksimum. Masing-masing dibulatkan dalam dua digit di belakang koma.

#### Contoh masukan dan keluaran

No	Masukan	Keluaran
1	60 30	3.06 318.13 45.92
2	100 45	7.22 1020.41 255.10
3	40 40	2.62 160.78 33.73