

Nama: Didik Weka Pratama

Kelas: IF-11-07

Nim: 2311102285

Kode soal: A

No 1. Source code:

```
package main
```

```
import (  
    "fmt"  
)
```

```
func hitungBiayaTiket(jumlahTiket int, jenisKursi string, isMember bool) int {  
    var hargaPerTiket int  
  
    if jenisKursi == "biasa" {  
        if isMember {  
            hargaPerTiket = 40000  
        } else {  
            hargaPerTiket = 50000  
        }  
    } else if jenisKursi == "VIP" {  
        if isMember {  
            hargaPerTiket = 60000  
        } else {  
            hargaPerTiket = 70000  
        }  
    } else {  
        fmt.Println("Jenis kursi tidak valid.")  
        return 0  
    }  
}
```

```

totalBiaya := jumlahTiket * hargaPerTiket

if jumlahTiket > 2 {
    totalBiaya = int(float64(totalBiaya) * 0.85)
}

return totalBiaya
}

func main() {
    var jumlahTiket int
    var jenisKursi string
    var isMember bool

    fmt.Print("Masukkan jumlah tiket: ")
    fmt.Scan(&jumlahTiket)

    fmt.Print("Masukkan jenis kursi (biasa/VIP): ")
    fmt.Scan(&jenisKursi)

    fmt.Print("Apakah Anda member? (true/false): ")
    fmt.Scan(&isMember)

    biaya := hitungBiayaTiket(jumlahTiket, jenisKursi, isMember)

    fmt.Printf("Total biaya tiket: Rp %d\n", biaya)
}

```

SS Output:

```
PS C:\golang> go run "c:\golang\src\hello.go"
Masukkan jumlah tiket: 1
Masukkan jenis kursi (biasa/VIP): VIP
Apakah Anda member? (true/false): true
Total biaya tiket: Rp 60000
PS C:\golang> 
```

Deskripsi program:

1. **Fungsi hitungBiayaTiket:** Fungsi ini menerima jumlah tiket, jenis kursi, dan status membership sebagai parameter. Ia menghitung total biaya berdasarkan tarif yang telah ditentukan.
2. **Harga Tiket:** Harga tiket ditentukan berdasarkan jenis kursi dan apakah penonton adalah member atau bukan.
3. **Diskon:** Jika jumlah tiket yang dibeli lebih dari 2, maka total biaya akan mendapatkan diskon 15%.
4. **Input dan Output:** Program meminta input dari pengguna dan menampilkan total biaya yang harus dibayar.

No 2. Source Code:

```
package main
```

```
import (  
    "fmt"  
)
```

```
func isBungaSempurna(jumlahKelopak int) bool {
```

```
    var totalKelopak int
```

```
    for i := 1; i < jumlahKelopak; i++ {
```

```
        if jumlahKelopak%i == 0 {
```

```
            totalKelopak += i
```

```
        }
```

```
    }
```

```
    return totalKelopak == jumlahKelopak
```

```
}
```

```
func findBungaSempurna(a, b int) []int {  
    var bungaSempurna []int  
  
    for i := a; i <= b; i++ {  
        if isBungaSempurna(i) {  
            bungaSempurna = append(bungaSempurna, i)  
        }  
    }  
  
    return bungaSempurna  
}
```

```
func main() {  
    var a, b int  
  
    fmt.Print("Masukkan jumlah kelopak (a): ")  
    fmt.Scan(&a)  
    fmt.Print("Masukkan jumlah kelopak (b): ")  
    fmt.Scan(&b)  
  
    if a <= 0 || b <= 0 || a > b {  
        fmt.Println("Input tidak valid. Pastikan a dan b adalah bilangan bulat positif dan a <= b.")  
        return  
    }  
  
    bungaSempurna := findBungaSempurna(a, b)  
  
    if len(bungaSempurna) > 0 {  
        fmt.Printf("Bunga sempurna antara %d dan %d: ", a, b)
```

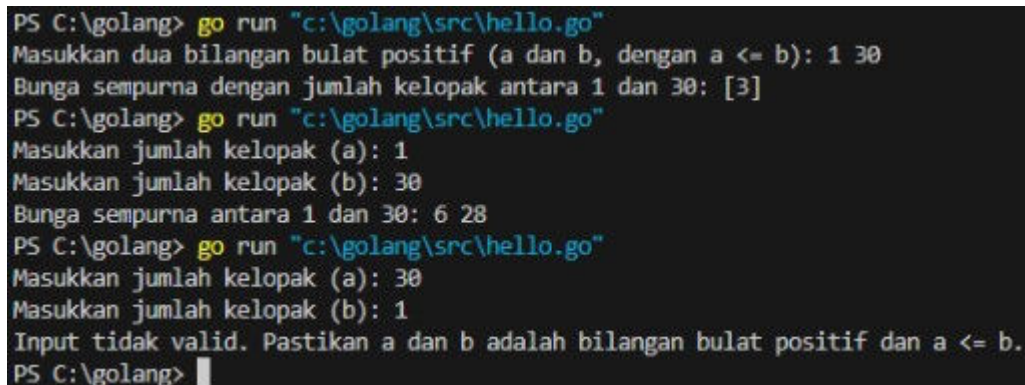
```

    for _, bunga := range bungaSempurna {
        fmt.Printf("%d ", bunga)
    }

    fmt.Println()
} else {
    fmt.Printf("Tidak ada bunga sempurna dengan jumlah kelopak antara %d dan %d.\n", a, b)
}
}

```

SS Output:



```

PS C:\golang> go run "c:\golang\src\hello.go"
Masukkan dua bilangan bulat positif (a dan b, dengan a <= b): 1 30
Bunga sempurna dengan jumlah kelopak antara 1 dan 30: [3]
PS C:\golang> go run "c:\golang\src\hello.go"
Masukkan jumlah kelopak (a): 1
Masukkan jumlah kelopak (b): 30
Bunga sempurna antara 1 dan 30: 6 28
PS C:\golang> go run "c:\golang\src\hello.go"
Masukkan jumlah kelopak (a): 30
Masukkan jumlah kelopak (b): 1
Input tidak valid. Pastikan a dan b adalah bilangan bulat positif dan a <= b.
PS C:\golang>

```

Deskripsi program:

1. **Fungsi isBungaSempurna:** Fungsi ini memeriksa apakah jumlah kelopak tertentu adalah bunga sempurna dengan menghitung jumlah faktor dari **jumlahKelopak** (tidak termasuk dirinya sendiri) dan membandingkannya dengan **jumlahKelopak**.
2. **Fungsi findBungaSempurna:** Fungsi ini mencari semua bunga sempurna dalam rentang dari **a** hingga **b** dan mengembalikan daftar bunga sempurna yang ditemukan.
3. **Input dan Validasi:** Program meminta pengguna untuk memasukkan dua bilangan bulat positif, **a** dan **b**, dan memvalidasi input untuk memastikan bahwa **a** dan **b** adalah bilangan positif dan **a <= b**.
4. **Output:** Program mencetak bunga sempurna yang ditemukan dalam rentang yang diberikan sesuai dengan format output yang diinginkan. Jika tidak ada bunga sempurna, program akan memberi tahu pengguna.

No 3. Source code:

```
package main
```

```
import (
```

```

    "fmt"
)

func hitungSesiPelatihan(hari, p, q int) int {

    if hari > 365 {
        return 0
    }

    if hari%p == 0 && hari%q != 0 {

        return 1 + hitungSesiPelatihan(hari+1, p, q)
    }

    return hitungSesiPelatihan(hari+1, p, q)
}

func main() {
    var p, q int

    fmt.Print("Masukkan kelipatan p (hari sesi pelatihan): ")
    fmt.Scan(&p)
    fmt.Print("Masukkan kelipatan q (hari tidak ada sesi pelatihan): ")
    fmt.Scan(&q)

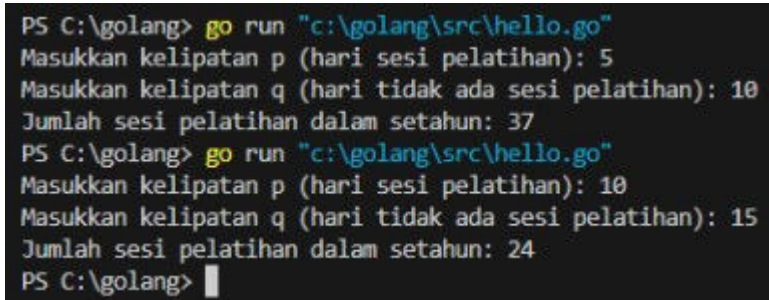
    if p <= 0 || q <= 0 {
        fmt.Println("Kelipatan p dan q harus bilangan bulat positif.")
        return
    }

    jumlahSesi := hitungSesiPelatihan(1, p, q)

```

```
    fmt.Printf("Jumlah sesi pelatihan dalam setahun: %d\n", jumlahSesi)
}
```

SS Output:



```
PS C:\golang> go run "c:\golang\src\hello.go"
Masukkan kelipatan p (hari sesi pelatihan): 5
Masukkan kelipatan q (hari tidak ada sesi pelatihan): 10
Jumlah sesi pelatihan dalam setahun: 37
PS C:\golang> go run "c:\golang\src\hello.go"
Masukkan kelipatan p (hari sesi pelatihan): 10
Masukkan kelipatan q (hari tidak ada sesi pelatihan): 15
Jumlah sesi pelatihan dalam setahun: 24
PS C:\golang> █
```

Deskripsi program:

1. **Fungsi hitungSesiPelatihan:** Fungsi ini adalah fungsi rekursif yang menghitung jumlah sesi pelatihan.
  - Basis: Jika **hari** lebih dari 365, maka tidak ada sesi pelatihan yang diadakan, sehingga fungsi mengembalikan 0.
  - Jika **hari** merupakan kelipatan **p** dan bukan kelipatan **q**, maka sesi pelatihan diadakan pada hari tersebut, dan fungsi mengembalikan 1 ditambah hasil dari pemanggilan rekursif untuk hari berikutnya.
  - Jika tidak, fungsi hanya melanjutkan ke hari berikutnya tanpa menambah jumlah sesi.
2. **Input dan Validasi:** Program meminta pengguna untuk memasukkan nilai **p** dan **q**, dan memvalidasi bahwa kedua nilai tersebut adalah bilangan bulat positif.
3. **Output:** Program mencetak jumlah sesi pelatihan yang diadakan dalam setahun berdasarkan aturan yang ditentukan.