

LATIHAN SOAL 2
ALGORITMA DAN PEMROGRAMAN 2



Oleh:

Muhammad Faris Rachmadi

103112400079

IF12-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

TIPE A

Soal 1 (Sewa Sepeda)

Code :

```
package main

import "fmt"

func hitungBiayaSewa(jam int, menit int, member bool, kodeVoucher
string) float64 {
    totalMenit := jam*60 + menit
    durasiJam := float64(totalMenit) / 60.0

    var biayaPerJam float64
    if member {
        biayaPerJam = 3500
    } else {
        biayaPerJam = 5000
    }

    biayaSewa := durasiJam * biayaPerJam
    if durasiJam > 3 {
        lastDigit := 0
        if len(kodeVoucher) > 0 {
            lastDigit = int(kodeVoucher[len(kodeVoucher)-1] - '0')
        }
        if lastDigit == 5 || lastDigit == 6 {
            biayaSewa *= 0.9
        }
    }

    return biayaSewa
}

func main() {
    var jam, menit int
    var isMemberStr string
    var isMember_13112400079 bool
    var kodeVoucher string

    fmt.Print("Masukkan durasi (jam): ")
    fmt.Scan(&jam)

    fmt.Print("Masukkan durasi (menit): ")
    fmt.Scan(&menit)
```

```

    fmt.Print("Apakah member? (true/false): ")
    fmt.Scan(&isMemberStr)
    if isMemberStr == "true" {
        isMember_13112400079 = true
    } else if isMemberStr == "false" {
        isMember_13112400079 = false
    }

    fmt.Print("Masukkan nomor voucher (jika ada): ")
    fmt.Scan(&kodeVoucher)

    biayaSetelahDiskon := hitungBiayaSewa(jam, menit,
isMember_13112400079, kodeVoucher)

    fmt.Printf("Biaya sewa setelah diskon (jika memenuhi syarat): Rp
%.2f\n", biayaSetelahDiskon)
}

```

Output:

```

PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\
Masukkan durasi (jam): 2
Masukkan durasi (menit): 30
Apakah member? (true/false): true
Masukkan nomor voucher (jika ada): 123456
Biaya sewa setelah diskon (jika memenuhi syarat): Rp 8750.00

```

Deskripsi:

Kode di atas adalah sebuah program Go yang menghitung biaya sewa berdasarkan durasi waktu, status keanggotaan, dan penggunaan voucher. Fungsi utama, `hitungBiayaSewa`, menerima input durasi waktu dalam jam dan menit, status keanggotaan, dan kode voucher. Jika pelanggan adalah anggota (member), tarif per jamnya lebih murah dibandingkan non-anggota. Selain itu, terdapat diskon 10% jika durasi sewa lebih dari tiga jam dan digit terakhir dari kode voucher adalah 5 atau 6. Fungsi main bertugas mengambil input dari pengguna, seperti durasi waktu, status keanggotaan, dan kode voucher, kemudian menghitung biaya sewa yang sudah dipotong diskon jika memenuhi syarat, dan menampilkan hasilnya ke layar.

Soal 2 (Perfect Number)

Code :

```
package main

import "fmt"

func jumlahFaktor(n int) int {
    jumlah := 1
    for i := 2; i*i <= n; i++ {
        if n%i == 0 {
            jumlah += i
            if i*i != n {
                jumlah += n / i
            }
        }
    }
    return jumlah
}

func main() {
    var a, b int

    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)

    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)

    fmt.Printf("Bilangan sempurna antara %d dan %d: ", a, b)
    ada := false
    for i := a; i <= b; i++ {
        if jumlahFaktor(i) == i && i > 1 {
            if ada {
                fmt.Print(", ")
            }
            fmt.Print(i)
            ada = true
        }
    }
    if !ada {
        fmt.Println("Tidak ada")
    } else {
        fmt.Println()
    }
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\
Masukkan nilai a: 3
Masukkan nilai b: 13
Bilangan sempurna antara 3 dan 13: 6
```

Deskripsi :

Kode program GO di atas bertujuan untuk menemukan bilangan sempurna dalam rentang yang ditentukan oleh pengguna (a sebagai batas bawah dan b sebagai batas atas). Fungsi `jumlahFaktor(n int)` digunakan untuk menghitung jumlah faktor pembagi dari bilangan `n`, dengan memperhitungkan faktor-faktor selain bilangan itu sendiri. Program utama mengambil input dari pengguna untuk nilai `a` dan `b`, kemudian memeriksa setiap bilangan dalam rentang tersebut apakah merupakan bilangan sempurna. Hasilnya, program akan mencetak daftar bilangan sempurna yang ditemukan, atau memberi tahu pengguna jika tidak ada bilangan sempurna di rentang tersebut.

Soal 3 (Rendezvous)

Code :

```
package main

import "fmt"

func main() {
    var x, y_103112400079 int

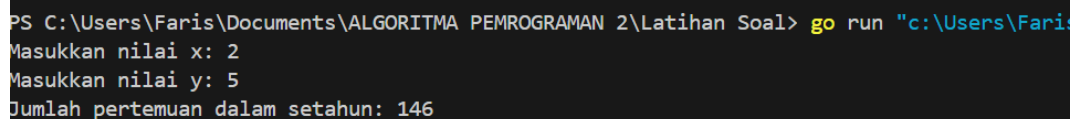
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)

    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y_103112400079)

    jumlahPertemuan := 0
    for hari := 1; hari <= 365; hari++ {
        if hari%x == 0 && hari%y_103112400079 != 0 {
            jumlahPertemuan++
        }
    }

    fmt.Printf("Jumlah pertemuan dalam setahun: %d\n",
jumlahPertemuan)
}
```

Output:



```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal\soal3.go"
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan dalam setahun: 146
```

Deskripsi :

Program GO di atas menghitung jumlah hari dalam setahun yang memenuhi kriteria tertentu berdasarkan dua angka yang dimasukkan pengguna, yaitu x dan y. Hari-hari yang dihitung adalah hari-hari yang merupakan kelipatan dari x namun bukan kelipatan dari y. Program menggunakan perulangan untuk memeriksa setiap hari dalam rentang 1 hingga 365 (satu tahun penuh) dan menambahkan ke hitungan total jika hari tersebut sesuai kriteria. Setelah loop selesai, program menampilkan jumlah total hari pertemuan tersebut ke layar.

TIPE B

Soal 1 (Voucher)

Code :

```
package main

import "fmt"

func main() {
    var a, b int

    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)

    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)

    jumlahGanjil := 0
    for i := a; i <= b; i++ {
        if i%2 != 0 {
            jumlahGanjil++
        }
    }

    fmt.Printf("Banyaknya angka ganjil: %d\n", jumlahGanjil)
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Fari
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500
```

Deskripsi :

Program Go di atas menghitung jumlah bilangan ganjil dalam sebuah rentang angka yang dimasukkan oleh pengguna. Program meminta pengguna untuk memberikan dua nilai integer, yaitu a sebagai batas bawah dan b sebagai batas atas. Dengan menggunakan perulangan dari a hingga b, program memeriksa setiap bilangan dalam rentang tersebut untuk menentukan apakah bilangan tersebut ganjil (bilangan dengan sisa hasil pembagian dengan 2 tidak sama dengan nol). Jika bilangan tersebut ganjil, maka program akan menambahkan jumlah ganjil yang ditemukan ke variabel jumlahGanjil_103112400079. Setelah proses perhitungan selesai, program menampilkan hasil berupa jumlah total bilangan ganjil dalam rentang yang diberikan oleh pengguna.

Soal 2 (Restoran)

Code :

```
package main

import "fmt"

func main() {
    var jumlahRombongan_103112400079 int
    fmt.Print("Masukkan jumlah rombongan: ")
    fmt.Scan(&jumlahRombongan_103112400079)

    for i := 1; i <= jumlahRombongan_103112400079; i++ {
        var menu, orang, sisa int
        fmt.Print("Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya): ")
        fmt.Scan(&menu, &orang, &sisa)

        var tarif int

        if menu <= 3 {
            tarif = 10000
        } else if menu <= 50 {
            tarif = 10000 + (menu-3)*2500
        } else {
            tarif = 100000
        }

        if sisa == 1 {
            tarif *= orang
        }

        fmt.Printf("Total biaya untuk rombongan %d : Rp %d\n", i, tarif)
    }
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\Documents\A
Masukkan jumlah rombongan: 2
Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya): 3 12 0
Total biaya untuk rombongan 1 : Rp 10000
Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya): 2 15 1
Total biaya untuk rombongan 2 : Rp 150000
```

Deskripsi :

Kode program GO di atas berfungsi untuk menghitung total biaya yang harus dibayar oleh beberapa rombongan berdasarkan jumlah menu makanan, jumlah orang, dan status sisa makanan. Pengguna diminta untuk memasukkan jumlah rombongan terlebih dahulu, lalu untuk setiap rombongan, pengguna memberikan detail jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak ada sisa, 1 jika ada sisa). Biaya dihitung berdasarkan jumlah menu makanan, dengan tarif tetap untuk menu sebanyak 3 atau kurang, tarif bertambah untuk jumlah menu lebih dari 3 hingga 50, dan tarif maksimum untuk lebih dari 50 menu. Jika makanan tersisa (status sisa = 1), tarif dikalikan dengan jumlah orang. Hasil akhirnya adalah total biaya untuk setiap rombongan, yang ditampilkan secara terpisah.

Soal 3 (Jumlah bilangan positif kelipatan 4)

Code :

```
package main

import "fmt"

func hitungKelipatan4Rekursif(jumlah_103112400079 int) int {
    var bilangan int
    fmt.Scan(&bilangan)

    if bilangan < 0 {
        return jumlah_103112400079
    }

    if bilangan > 0 && bilangan%4 == 0 {
        jumlah_103112400079 += bilangan
    }
    return hitungKelipatan4Rekursif(jumlah_103112400079)
}

func main() {
    fmt.Println("Masukkan bilangan (negatif untuk berhenti):")
    jumlahKelipatan := hitungKelipatan4Rekursif(0)
    fmt.Printf("Jumlah bilangan kelipatan 4: %d\n", jumlahKelipatan)
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Jumlah bilangan kelipatan 4: 4
```

Deskripsi :

Program di atas menggunakan pendekatan rekursif untuk menghitung jumlah bilangan kelipatan 4 yang dimasukkan oleh pengguna. Fungsi `hitungKelipatan4Rekursif` menerima nilai awal `jumlah_103112400079` yang akan ditambahkan dengan bilangan kelipatan 4 selama proses rekursi. Pengguna dapat memasukkan bilangan satu per satu, dan proses akan berhenti ketika bilangan negatif dimasukkan. Jika bilangan yang diberikan adalah kelipatan 4, maka nilainya ditambahkan ke total jumlah. Fungsi tersebut dipanggil kembali secara rekursif hingga kondisi penghentian (bilangan negatif) terpenuhi. Hasil akhirnya adalah jumlah total bilangan kelipatan 4 yang dicetak oleh program.

TIPE C

Soal 1 (Digit)

Code :

```
package main

import "fmt"

func main() {
    var n_103112400079 int
    fmt.Print("Masukkan bilangan bulat positif (>10): ")
    fmt.Scan(&n_103112400079)

    panjang := 0
    temp := n_103112400079
    for temp > 0 {
        temp /= 10
        panjang++
    }

    tengah := panjang / 2
    pembagi := 1
    for i := 0; i < panjang-(tengah+(panjang%2)); i++ {
        pembagi *= 10
    }

    n1 := n_103112400079 / pembagi
    n2 := n_103112400079 % pembagi

    fmt.Println("Bilangan 1:", n1)
    fmt.Println("Bilangan 2:", n2)
    fmt.Println("Hasil penjumlahan:", n1+n2)
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168
```

Deskripsi :

Program di atas bertujuan untuk membagi bilangan bulat positif yang memiliki lebih dari satu digit menjadi dua bagian dan kemudian menghitung hasil penjumlahan dari kedua bagian tersebut. Pertama, pengguna memasukkan bilangan bulat positif yang lebih besar dari 10. Program menghitung jumlah digit dalam bilangan tersebut dan menentukan posisi pembagian di bagian tengah. Setelah itu, bilangan dibagi menjadi dua bagian, $n1$ dan $n2$, berdasarkan posisi pembagian. Bagian pertama ($n1$) diperoleh dengan membagi bilangan dengan pembagi, sedangkan bagian kedua ($n2$) adalah hasil modulus dari bilangan terhadap pembagi. Hasilnya adalah kedua bagian serta penjumlahan antara $n1$ dan $n2$ yang ditampilkan kepada pengguna.

Soal 2 (Bukber IF)

Code :

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&n)

    var hadiahA, hadiahB, hadiahC_103112400079 int
    for i := 0; i < n; i++ {
        var nomorKartu int
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i+1)
        fmt.Scan(&nomorKartu)

        // Memeriksa jika semua digit sama (Hadiah A)
        if nomorKartu/100 == nomorKartu%10 && (nomorKartu/10)%10
        == nomorKartu%10 {
            fmt.Println("Hadiah A")
            hadiahA++
        } else if isDifferentDigits(nomorKartu) {
            // Memeriksa jika semua digit berbeda (Hadiah B)
            fmt.Println("Hadiah B")
            hadiahB++
        } else {
            // Hadiah C untuk kombinasi angka
            fmt.Println("Hadiah C")
            hadiahC_103112400079++
        }
    }

    // Output jumlah hadiah yang diberikan
    fmt.Printf("\nJumlah yang memperoleh Hadiah A: %d\n", hadiahA)
    fmt.Printf("Jumlah yang memperoleh Hadiah B: %d\n", hadiahB)
    fmt.Printf("Jumlah yang memperoleh Hadiah C: %d\n",
    hadiahC_103112400079)
}

// Fungsi untuk memeriksa apakah semua digit berbeda
func isDifferentDigits(nomorKartu int) bool {
    digit1 := nomorKartu / 100
    digit2 := (nomorKartu / 10) % 10
    digit3 := nomorKartu % 10
```

```
    return digit1 != digit2 && digit1 != digit3 && digit2 != digit3
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\
Masukkan jumlah peserta: 3
Masukkan nomor kartu peserta ke-1: 333
Hadiah A
Masukkan nomor kartu peserta ke-2: 123
Hadiah B
Masukkan nomor kartu peserta ke-3: 898
Hadiah C

Jumlah yang memperoleh Hadiah A: 1
Jumlah yang memperoleh Hadiah B: 1
Jumlah yang memperoleh Hadiah C: 1
```

Deskripsi :

Program di atas menghitung distribusi hadiah berdasarkan nomor kartu peserta yang dimasukkan. Pengguna diminta untuk memasukkan jumlah peserta, lalu setiap peserta memberikan nomor kartu tiga digit. Hadiah diberikan berdasarkan kriteria tertentu: jika semua digit pada nomor kartu sama, peserta mendapatkan *Hadiah A*; jika semua digit berbeda, peserta mendapatkan *Hadiah B*; sedangkan untuk kombinasi angka lainnya, peserta menerima *Hadiah C*. Program menggunakan fungsi `isDifferentDigits` untuk memeriksa apakah semua digit pada nomor kartu berbeda. Setelah memproses semua peserta, program mencetak jumlah total penerima untuk masing-masing jenis hadiah.

Soal 3 (Perkalian dengan cara penjumlahan)

Code :

```
package main

import "fmt"

func perkalian(n, m, hasil int) int {
    if m == 0 {
        return hasil
    }
    return perkalian(n, m-1, hasil+n)
}

func main() {
    var n, m_103112400079 int

    fmt.Print("Masukkan bilangan n: ")
    fmt.Scanln(&n)

    fmt.Print("Masukkan bilangan m: ")
    fmt.Scanln(&m_103112400079)

    hasil := perkalian(n, m_103112400079, 0)
    fmt.Printf("Hasil dari %d x %d = %d\n", n, m_103112400079, hasil)
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Latihan Soal> go run "c:\Users\Faris\
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Hasil dari 5 x 6 = 30
```

Deskripsi :

Program ini menggunakan rekursi untuk menghitung hasil perkalian dua bilangan bulat positif yang dimasukkan oleh pengguna. Fungsi perkalian mengembalikan hasil penjumlahan n secara berulang sebanyak m kali hingga m mencapai nol sebagai kondisi penghentian. Fungsi ini memanfaatkan parameter hasil sebagai akumulator untuk menyimpan hasil sementara selama rekursi berlangsung. Di dalam fungsi utama (main), pengguna diminta untuk memasukkan nilai n dan m , yang kemudian diproses oleh fungsi rekursif untuk menghasilkan hasil perkalian. Setelah proses selesai, hasil perkalian ditampilkan kepada pengguna melalui layar.