

LATIHAN SOAL 2
ALGORITMA DAN PEMROGRAMAN 2
SOAL TIPE A, SOAL TIPE B, SOAL TIPE C



Oleh:

FEROS PEDROSA VALENTINO

103112400055

IF-12-01

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2025

SOAL TIPE A

1. Soal 1 (Sewa Sepeda)

Source code:

```
package main

import "fmt"

func hitungBiayaSewa(jam int, menit int, isMember bool,
    nomorVoucher string) float64 {
    const tarifNonMember = 5000.0
    const tarifMember = 3500.0
    const diskonPersen = 0.10

    totalMenit := jam*60 + menit
    if totalMenit < 60 {
        return 0.0
    }

    var tarif float64
    if isMember {
        tarif = tarifMember
    } else {
        tarif = tarifNonMember
    }

    durasiSewa := totalMenit / 60
    if totalMenit%60 > 0 {
        durasiSewa++
    }

    biayaSewa := float64(durasiSewa) * tarif

    if len(nomorVoucher) >= 5 && len(nomorVoucher) <= 6
    {
        biayaSewa = biayaSewa * (1 - diskonPersen)
    }

    return biayaSewa
}

func main() {
    var jam_103112400055, menit int
    var isMember bool
    var nomorVoucher string
```

```

// Input pengguna
fmt.Print("Masukkan durasi (jam): ")
fmt.Scan(&jam_103112400055)
fmt.Print("Masukkan durasi (menit): ")
fmt.Scan(&menit)
fmt.Print("Apakah member? (true/false): ")
fmt.Scan(&isMember)

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

biaya := hitungBiayaSewa(jam_103112400055, menit,
isMember, nomorVoucher)
fmt.Printf("Biaya sewa setelah diskon (jika memenuhi
syarat): Rp %.2f\n", biaya)
}

```

Output:

```

PS C:\ALPRO\semester2_alpro2\SoalTipeA> go run "c:\ALPRO\semester2_alpro2\SoalTipeA\tipeA_1\SewaSepeda.go"
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 9450.00

```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menghitung biaya sewa berdasarkan durasi waktu, keanggotaan member, penggunaan voucher. Fungsi hitungBiayaSewa menerima beberapa masukan, antara lain jam, menit, status keanggotaan (apakah anggota atau non-anggota), serta nomor voucher. Tarif sewa terbagi menjadi dua, yakni Rp 3. 500 per jam untuk member dan Rp 5. 000 per jam untuk nonmember. Jika durasi sewa kurang dari satu jam, maka biaya sewa akan dianggap nol. Durasi sewa dihitung dalam satuan jam dengan pembulatan untuk menit tambahan. Jika nomor voucher yang dimiliki memenuhi syarat panjang tertentu, pengguna akan berhak mendapatkan diskon sebesar 10%. Program ini meminta inputan dari pengguna untuk menentukan lama sewa dalam durasi jam dan menit, status member, dan nomor voucher(jika ada), sebelum akhirnya mencetak total biaya sewa yang harus dibayar.

2. Soal 2 (Perfect Number)

Source code:

```
package main

import "fmt"

func isPerfectNumber(n int) bool {
    sum := 0
    for i := 1; i <= n/2; i++ {
        if n%i == 0 {
            sum += i
        }
    }
    return sum == n
}

func main() {
    var a_103112400055, b int

    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a_103112400055)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)

    fmt.Printf("Perfect numbers antara %d dan %d: ",
a_103112400055, b)
    for i := a_103112400055; i <= b; i++ {
        if isPerfectNumber(i) {
            fmt.Printf("%d ", i)
        }
    }
    fmt.Println()
}
```

Output:

```
PS C:\ALPRO\semester2_alpro2\SoalTipeA> go run "c:\ALPRO\semester2_alpro2\SoalTipeA\tipeA_2\PerfectNumber.go"
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6
```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menghitung perfect numbers antara dua bilangan yang diinputkan pengguna. Fungsi isPerfectNumber akan memeriksa setiap angka dalam rentang kedua angka yang diinputkan untuk menentukan apakah angka tersebut adalah bilangan sempurna, yaitu angka yang sama dengan jumlah faktor-faktornya (kecuali dirinya sendiri). Setelah

proses pemeriksaan selesai, program akan menampilkan semua bilangan sempurna yang ditemukan di antara nilai-nilai yang telah dimasukkan. Di dalam fungsi main pengguna diminta untuk memasukkan dua nilai bulat positif yang akan menjadi batas bawah dan batas atas dari rentang yang akan dicari.

3. Soal 3 (Rendezvous)

Source code:

```
package main

import "fmt"

func main() {
    var x_103112400055, y int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x_103112400055)
    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y)

    jumlahPertemuan := 0
    for hari := 1; hari <= 365; hari++ {
        if hari%x_103112400055 == 0 && hari%y != 0 {
            jumlahPertemuan++
        }
    }

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

Output:

```
PS C:\ALPRO\semester2_alpro2\SoalTipeA> go run "c:\ALPRO\semester2_alpro2\SoalTipeA\tipeA_3\Rendezvous.go"
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan dalam setahun: 146
```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menghitung jumlah hari pertemuan rahasia antara dua agen mata-mata selama satu tahun. Para agen mata-mata ini akan berjumpa pada hari-hari yang merupakan kelipatan dari bilangan positif x, namun tidak termasuk kelipatan dari bilangan positif y. Pengguna diminta untuk memasukkan nilai x dan y, setelah itu program akan menghitung secara iteratif jumlah hari di mana pertemuan dapat berlangsung. Hasil akhir dari perhitungan ini akan ditampilkan kepada pengguna sebagai jumlah pertemuan dalam setahun.

SOAL TIPE B

1. Soal 1 (Voucher)

Source code:

```
package main

import "fmt"

func main() {
    var a_103112400055, b int

    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a_103112400055)

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

    count := 0
    for i := a_103112400055; i <= b; i++ {
        if i%2 != 0 {
            count++
        }
    }

    fmt.Println("Banyaknya angka ganjil:", count)
}
```

Output:

```
PS C:\ALPRO\semester2_alpro2\LATIHAN SOAL 2> go run "c:\ALPRO\semester2_alpro2\LATIHAN SOAL 2\SoalTipeB\tipeB_1\voucher.go"
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500
```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menghitung jumlah bilangan ganjil dalam rentang kedua bilangan yang diinputkan. Pertama deklarasikan variabel `a_103112400055`, `b` sebagai tipe data integer. Lalu program meminta pengguna untuk memasukkan nilai `a` dan nilai `b`. Setelah menerima input, program melakukan iterasi dari angka `a` hingga `b` untuk memeriksa setiap angka, apakah angka tersebut merupakan bilangan ganjil (memiliki sisa 1 ketika dibagi 2). Saat program menemukan bilangan ganjil, `count` akan ditambahkan. Pada akhir proses, program akan mencetak jumlah total bilangan ganjil yang ditemukan dalam rentang kedua nilai yang diinputkan.

2. Soal 2 (Restoran)

Source code:

```
package main

import "fmt"

func main() {
    var rombongan_103112400055 int
    fmt.Print("Masukkan jumlah rombongan: ")
    fmt.Scan(&rombongan_103112400055)

    for i := 0; i < rombongan_103112400055; i++ {
        var menu, orang int
        var sisa bool

        fmt.Printf("Masukkan jumlah menu, jumlah orang, dan
        status sisa makanan (0 untuk tidak, 1 untuk iya): ")
        fmt.Scan(&menu, &orang, &sisa)

        totalBiaya := 10000
        if menu > 3 {
            totalBiaya += (menu - 3) * 2500
        }
        if menu > 50 {
            totalBiaya = 100000
        }

        if sisa {
            totalBiaya *= orang
        }

        fmt.Printf("Total biaya untuk rombongan %d: Rp %d\n",
        i+1, totalBiaya)
    }
}
```

Output:

```
PS C:\ALPRO\semester2_alpro2\LATIHAN SOAL 2> go run "c:\ALPRO\semester2_alpro2\LATIHAN SOAL 2\SoalTipeB\tipe8_2\Restoran.go"
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 program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menghitung total biaya rombongan yang memesan menu makanan. Pertama deklarasikan variabel rombongan_103112400055 sebagai tipe data integer. Lalu program meminta untuk memasukkan jumlah rombongan. Untuk

setiap rombongan, pengguna perlu memasukkan jumlah menu yang dipesan, jumlah orang dalam rombongan, serta status mengenai apakah terdapat sisa makanan. Biaya dasar ditetapkan sebesar Rp 10.000, dengan tambahan biaya jika jumlah menu yang dipesan melebihi tiga, dan terdapat batasan maksimum biaya sebesar Rp 100.000 jika jumlah menu lebih dari 50. Jika ada sisa makanan, total biaya akan dikalikan dengan jumlah orang dalam rombongan. Terakhir, program akan mencetak total biaya untuk setiap rombongan.

3. Soal 3 (Jumlah bilangan positif kelipatan 4)

Source code:

```
package main

import "fmt"

// Fungsi rekursif untuk menjumlahkan bilangan kelipatan 4
func jumlahKelipatanEmpat(numbers []int, index int) int {
    if index == len(numbers) {
        return 0
    }

    if numbers[index] > 0 && numbers[index]%4 == 0 {
        return numbers[index] + jumlahKelipatanEmpat(numbers,
index+1)
    }

    return jumlahKelipatanEmpat(numbers, index+1)
}

func main() {
    var numbers []int
    var input_103112400055 int

    fmt.Println("Masukkan bilangan (negatif untuk berhenti): ")

    for {
        fmt.Scan(&input_103112400055)
        if input_103112400055 < 0 {
            break
        }
        numbers = append(numbers, input_103112400055)
    }

    total := jumlahKelipatanEmpat(numbers, 0)

    fmt.Printf("Jumlah bilangan kelipatan 4: %d\n", total)
}
```

Output:

```
PS C:\ALPRO\semester2_alpro2\LATIHAN SOAL 2> go run "c:\ALPRO\semester2_alpro2\LATIHAN SOAL 2\SoalTipeB\tipeB_3\jmlhbilpstif.go"
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Jumlah bilangan kelipatan 4: 4
```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menjumlahkan bilangan bulat positif yang kelipatannya 4. Program ini menerima input dari pengguna berupa deretan bilangan bulat, dengan

bilangan negatif sebagai tanda untuk berhenti. Menggunakan algoritma rekursif, fungsi `jumlahKelipatanEmpat` akan memeriksa setiap elemen dalam array untuk menentukan apakah elemen tersebut adalah positif dan merupakan kelipatan 4. Jika elemen tersebut memenuhi kriteria, maka nilainya akan ditambahkan ke total. Setelah pengguna selesai memasukkan bilangan, program akan mencetak hasil akhir, yaitu jumlah dari semua bilangan yang merupakan kelipatan 4.

SOAL TIPE C

1. Soal 1 (Digit)

Source code:

```
package main

import "fmt"

func main() {
    var number_103112400055 int
    fmt.Print("Masukkan bilangan bulat positif (>10): ")
    fmt.Scan(&number_103112400055)

    if number_103112400055 <= 10 {
        fmt.Println("Silakan masukkan bilangan lebih besar dari 10.")
        return
    }

    digits := []int{}
    temp := number_103112400055
    for temp > 0 {
        digits = append([]int{temp % 10}, digits...)
        temp /= 10
    }

    length := len(digits)
    var bilangan1, bilangan2 int

    if length%2 == 0 {
        mid := length / 2
        for i := 0; i < mid; i++ {
            bilangan1 = bilangan1*10 + digits[i]
        }
        for i := mid; i < length; i++ {
            bilangan2 = bilangan2*10 + digits[i]
        }
    } else {
        mid := length / 2
        for i := 0; i <= mid; i++ {
            bilangan1 = bilangan1*10 + digits[i]
        }
        for i := mid + 1; i < length; i++ {
            bilangan2 = bilangan2*10 + digits[i]
        }
    }
}
```

```

    }

    hasilPenjumlahan := bilangan1 + bilangan2

    fmt.Printf("Bilangan 1: %d\n", bilangan1)
    fmt.Printf("Bilangan 2: %d\n", bilangan2)
    fmt.Printf("Hasil penjumlahan: %d\n", hasilPenjumlahan)
}

```

Output:

```

PS C:\ALPRO\semester2_alpro2\LATIHAN SOAL 2> go run "c:\ALPRO\semester2_alpro2\LATIHAN SOAL 2\SoalTipeC\tipeC_1\digit.go"
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6
PS C:\ALPRO\semester2_alpro2\LATIHAN SOAL 2> go run "c:\ALPRO\semester2_alpro2\LATIHAN SOAL 2\SoalTipeC\tipeC_1\digit.go"
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168

```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk memotong suatu bilangan bulat positif menjadi dua bagian yang sama panjang. Jika jumlah digit bilangan ganjil, bagian kiri akan mencakup satu digit lebih banyak dibandingkan bagian kanan. Pengguna diminta untuk memasukkan bilangan bulat positif lebih besar dari 10. Jika bilangan yang dimasukkan tidak memenuhi syarat, program akan memberikan pesan “Silakan masukkan bilangan lebih besar dari 10.”. Bilangan tersebut kemudian dipecah menjadi digit-digit terpisah sebelum dipisahkan menjadi dua bilangan baru. Jika panjang digit dari bilangan genap, kedua bilangan akan dibuat dari separuh digit tersebut. Namun, jika panjang digit adalah ganjil, bilangan pertama akan mengambil satu digit tambahan di kirinya. Terakhir, program akan menghitung dan menampilkan kedua bilangan hasil pemotongan serta hasil penjumlahan dari keduanya.

2. Soal 2 (Bukber IF)

Source code:

```
package main

import "fmt"

func main() {
    var n_103112400055 int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&n_103112400055)

    countA, countB, countC := 0, 0, 0

    for i := 1; i <= n_103112400055; i++ {
        var number int
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i)
        fmt.Scan(&number)

        numStr := fmt.Sprintf("%d", number)
        if allSame(numStr) {
            fmt.Println("Hadiah A")
            countA++
        } else if allDifferent(numStr) {
            fmt.Println("Hadiah B")
            countB++
        } else {
            fmt.Println("Hadiah C")
            countC++
        }
    }

    fmt.Println("Jumlah yang memperoleh Hadiah A:",
countA)
    fmt.Println("Jumlah yang memperoleh Hadiah B:",
countB)
    fmt.Println("Jumlah yang memperoleh Hadiah C:",
countC)
}

func allSame(s string) bool {
    for i := 1; i < len(s); i++ {
        if s[i] != s[0] {
            return false
        }
    }
    return true
}
```

```

    }

    func allDifferent(s string) bool {
        seen := make(map[rune]bool)
        for _, ch := range s {
            if seen[ch] {
                return false
            }
            seen[ch] = true
        }
        return true
    }
}

```

Output:

```

PS C:\ALPRO\semester2_alpro2\LATIHAN SOAL 2> go run "c:\ALPRO\semester2_alpro2\LATIHAN SOAL 2\SoalTipeC\tipeC_2\bukber.go"
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:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menentukan hadiah yang diperoleh peserta lomba berdasarkan nomor kartu peserta yang dimasukkan. Program ini menerima jumlah peserta dan nomor kartu peserta, kemudian menganalisis nomor-nomor tersebut sesuai dengan kriteria yang telah ditetapkan. Dalam setiap iterasi, program meminta input berupa nomor kartu peserta, lalu mengonversinya menjadi string untuk analisis digit. Terdapat dua fungsi pembantu yang bernama allSame dan allDifferent. Fungsi allSame berfungsi untuk memeriksa apakah semua digit dalam nomor kartu adalah sama, sementara allDifferent memeriksa apakah semua digit dalam nomor kartu tersebut berbeda satu sama lain. Apabila nomor kartu peserta memiliki semua digit yang sama, peserta akan menerima “Hadiah A”. Jika semua digit yang ada berbeda, peserta akan mendapatkan “Hadiah B”. Namun, jika nomor kartu tidak termasuk dalam kedua kategori tersebut, peserta akan memperoleh “Hadiah C”. Setelah memasukkan semua nomor kartu, program akan menampilkan hasil berupa jenis hadiah yang diperoleh setiap peserta dan jumlah peserta yang mendapatkan masing-masing hadiah.

3. Soal 3 (Perkalian dengan cara penjumlahan)

Source code:

```
package main

import "fmt"

// Fungsi untuk menghitung perkalian secara rekursif
func multiply(n int, m int) int {
    if m == 0 {
        return 0
    }
    return n + multiply(n, m-1)
}

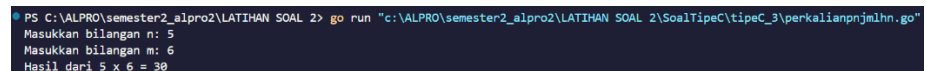
func main() {
    var n_103112400055, m int

    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&n_103112400055)
    fmt.Print("Masukkan bilangan m: ")
    fmt.Scan(&m)

    hasil := multiply(n_103112400055, m)

    fmt.Printf("Hasil dari %d x %d = %d\n",
n_103112400055, m, hasil)
}
```

Output:



```
PS C:\ALPRO\semester2_alpro2\LATIHAN SOAL 2> go run "c:\ALPRO\semester2_alpro2\LATIHAN SOAL 2\SoalTipeC\tipeC_3\perkalianpnjmlhn.go"
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Hasil dari 5 x 6 = 30
```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk menghitung hasil perkalian kedua bilangan yang diinputkan. Fungsi multiply berguna dalam menjalankan operasi ini, di mana nilai m bernilai 0, fungsi tersebut akan mengembalikan 0, sesuai dengan prinsip bahwa perkalian dengan angka nol selalu menghasilkan nol. Jika m bukan nol, program akan melanjutkan dengan menambahkan nilai n ke hasil pemanggilan fungsi multiply secara rekursif, sambil mengurangi nilai m sebanyak 1. Dalam fungsi utama, pengguna akan diminta untuk memasukkan dua bilangan bulat, yaitu n dan m. Setelah itu, program akan menghitung hasil perkalian dengan memanggil fungsi multiply dan menampilkan hasilnya.