

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

Latihan Soal 2

MATERI



Oleh:

MUHAMMAD ZAKY MUBAROK

103112400073

KELAS

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

SOAL TIPE A :

1. Sewa Sepeda

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jam, menit int
    var anggota_103112400073 bool
    var kupon string
    var totalBiaya float64

    fmt.Print("Masukkan durasi (jam): ")
    fmt.Scan(&jam)
    fmt.Print("Masukkan durasi (menit): ")
    fmt.Scan(&menit)
    fmt.Print("Apakah member? (true/false): ")
    fmt.Scan(&anggota_103112400073)
    fmt.Print("Masukkan nomor kupon (jika ada): ")
    fmt.Scan(&kupon)

    totalWaktu := float64(jam) + float64(menit)/60

    var tarifPerJam float64
    if anggota_103112400073 {
        tarifPerJam = 3500
    } else {
        tarifPerJam = 5000
    }

    totalBiaya = totalWaktu * tarifPerJam

    if totalWaktu > 3 {
        totalBiaya *= 0.9
    }
}
```

```

    if len(kupon) == 5 {
        totalBiaya *= 0.9
    }

    totalBiaya = math.Round(totalBiaya*100) / 100
    fmt.Printf("Biaya sewa setelah diskon (jika memenuhi syarat): Rp
%.2f\n", totalBiaya)
}

```

```

PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2> go run "d:\ALPRO 2\103
a.go"
Masukkan durasi (jam): 2
Masukkan durasi (menit): 30
Apakah member? (true/false): true
Masukkan nomor kupon (jika ada): 123456
Biaya sewa setelah diskon (jika memenuhi syarat): Rp 8750.00
PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2>

```

Penjelasan :

Kode ini menghitung biaya sewa berdasarkan durasi pemakaian, status keanggotaan, dan penggunaan kupon diskon. Berikut alurnya:

1. **Input Data** – Pengguna memasukkan durasi dalam jam & menit, status keanggotaan, dan nomor kupon (jika ada).
2. **Konversi Waktu** – Durasi dihitung dalam bentuk desimal.
3. **Penentuan Tarif** – Jika pengguna adalah anggota, tarifnya Rp 3.500 per jam, jika bukan, Rp 5.000 per jam.
4. **Penghitungan Biaya** – Total biaya didapat dari tarif per jam dikali jumlah jam.
5. **Diskon:**
 - Jika durasi lebih dari 3 jam, diberikan diskon 10%.
 - Jika kupon memiliki tepat 5 karakter, diberikan diskon tambahan 10%.
6. **Pembulatan Biaya** – Hasil akhir dibulatkan hingga dua desimal untuk presisi.
7. **Output** – Program mencetak biaya akhir setelah diskon.

2. Perfect Number

```
package main

import "fmt"

func PerfectNumber(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_10311240073, b int

    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a_10311240073)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)

    fmt.Printf("Perfect numbers antara %d dan %d: ", a_10311240073,
b)
    for i := a_10311240073; i <= b; i++ {
        if PerfectNumber(i) {
            fmt.Print(i, " ")
        }
    }
    fmt.Println()
}
```

```
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6
```

Penjelasan :

Kode ini mencari dan mencetak **bilangan sempurna** dalam rentang yang diberikan pengguna. Berikut alurnya:

1. **Fungsi `PerfectNumber`** – Mengecek apakah suatu bilangan adalah **bilangan sempurna** dengan menjumlahkan faktor-faktornya dan membandingkannya dengan bilangan itu sendiri.
2. **Input Data** – Pengguna memasukkan dua batas angka (`a_103112400073` dan `b`).
3. **Iterasi & Pengecekan** – Program memeriksa setiap angka dalam rentang tersebut menggunakan fungsi `PerfectNumber`.
4. **Output** – Bilangan sempurna yang ditemukan dicetak ke layar.

3. Rendezvous

```
package main

import "fmt"

func HitungKetemu(x, y int) int {
    hitung := 0
    for day := 1; day <= 365; day++ {

        if day%x == 0 && day%y != 0 {
            hitung++
        }
    }
    return hitung
}

func main() {
    var x_103112400073, y int

    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x_103112400073)
    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y)

    PertemuanRahasia := HitungKetemu(x_103112400073, y)

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

```
PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2> g
.go"
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan dalam setahun: 146
```

Penjelasan :

Kode ini menghitung jumlah pertemuan dalam setahun berdasarkan dua bilangan (x dan y) dengan logika berikut:

1. **Fungsi HitungKetemu** – Menghitung hari-hari dalam satu tahun (365 hari) yang merupakan kelipatan x , tetapi bukan kelipatan y .
2. **Input Data** – Pengguna memasukkan dua bilangan ($x_{_103112400073}$ dan y).
3. **Iterasi & Pengecekan** – Program memeriksa setiap hari dalam setahun apakah memenuhi kondisi kelipatan x dan bukan kelipatan y .
4. **Output** – Menampilkan jumlah hari yang memenuhi kondisi tersebut.

SOAL TIPE B :

1. Voucher

```
package main

import "fmt"

func hitungAngkaGanjil(a, b int) int {
    hitung := 0
    for i := a; i <= b; i++ {
        if i%2 != 0 {
            hitung++
        }
    }
    return hitung
}

func main() {
    var a_103112400073, b int

    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a_103112400073)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)

    angkaGanjil := hitungAngkaGanjil(a_103112400073, b)

    fmt.Print("Banyaknya angka ganjil : " , angkaGanjil)
}
```

```
PS D:\ALPRO 2\103112400073_LATIHAN>
"
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil : 500
```


Penjelasan :

Kode ini menghitung jumlah **bilangan ganjil** dalam rentang tertentu dengan langkah-langkah berikut:

1. **Fungsi hitungAngkaGanjil** – Memeriksa setiap angka dalam rentang $[a, b]$ dan menghitung berapa banyak yang bernilai ganjil.
2. **Input Data** – Pengguna memasukkan dua batas angka ($a_103112400073$ dan b).
3. **Iterasi & Pengecekan** – Program memeriksa setiap angka dalam rentang tersebut dan menambah hitungan jika angka tersebut ganjil ($i \% 2 \neq 0$).
4. **Output** – Menampilkan jumlah bilangan ganjil yang ditemukan.

2. Restoran

```
package main

import "fmt"

func hitungBiaya(jumlahMenu, jumlahOrang int, adaSisa bool) int {
    var biayaPerMenu int
    var totalBiaya_103112400073 int

    if jumlahMenu <= 3 {
        biayaPerMenu = 10000
    } else if jumlahMenu > 50 {
        biayaPerMenu = 100000
    } else {
        biayaPerMenu = 10000 + (2500 * (jumlahMenu - 3))
    }

    totalBiaya_103112400073 = biayaPerMenu

    if adaSisa {
        totalBiaya_103112400073 *= jumlahOrang
    }

    return totalBiaya_103112400073
}

func main() {
    var jumlahRombongan int

    fmt.Print("Masukkan jumlah rombongan: ")
    fmt.Scan(&jumlahRombongan)

    for i := 1; i <= jumlahRombongan; i++ {
        var jumlahMenu, jumlahOrang int
        var sisaMakanan int
    }
}
```

```

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

        adaSisa := sisaMakanan == 1

        totalBiaya := hitungBiaya(jumlahMenu, jumlahOrang, adaSisa)

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

```

```

PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2> go run "d:\ALPRO 2\103112400073_LATIHAN SOAL 2\Soal tipe
o"
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
PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2>

```

Penjelasan :

Kode ini menghitung **total biaya makanan** untuk beberapa rombongan berdasarkan jumlah menu, jumlah orang, dan status sisa makanan. Berikut alur kerjanya:

1. **Fungsi hitungBiaya:**
 - Menentukan **biaya per menu**:
 - ≤ 3 menu: Rp 10.000.
 - 50 menu: Rp 100.000.
 - 4–50 menu: Biaya dasar Rp 10.000 ditambah Rp 2.500 per menu tambahan.
 - Jika ada sisa makanan, biaya dihitung untuk setiap orang dalam rombongan.
2. **Input Data:**
 - Pengguna memasukkan jumlah rombongan, lalu untuk masing-masing rombongan, diminta jumlah menu, jumlah orang, dan status sisa makanan (0 atau 1).
3. **Iterasi:**

- Untuk setiap rombongan, fungsi menghitung biaya berdasarkan input, lalu mencetak total biaya.

4. **Output:**

- Total biaya untuk masing-masing rombongan ditampilkan.

3. Bilangan Positif Kelipatan

```
package main

import "fmt"

func jumlahKelipatan4(total int) int {
    var bilangan_10311240073 int
    fmt.Print("Masukkan bilangan (negatif untuk berhenti): ")
    fmt.Scan(&bilangan_10311240073)

    if bilangan_10311240073 < 0 {
        return total
    }

    if bilangan_10311240073 > 0 && bilangan_10311240073%4 == 0 {
        total += bilangan_10311240073
    }

    return jumlahKelipatan4(total)
}

func main() {
    total := jumlahKelipatan4(0)
    fmt.Printf("Jumlah bilangan kelipatan 4 : %d\n", total)
}
```

```
PS D:\ALPRO 2\10311240073_LATIHAN SOAL 2> go run
f kelipatan 4.go
Masukkan bilangan (negatif untuk berhenti): 2
Masukkan bilangan (negatif untuk berhenti): 3
Masukkan bilangan (negatif untuk berhenti): 4
Masukkan bilangan (negatif untuk berhenti): 5
Masukkan bilangan (negatif untuk berhenti): 6
Masukkan bilangan (negatif untuk berhenti): -1
Jumlah bilangan kelipatan 4 : 4
PS D:\ALPRO 2\10311240073_LATIHAN SOAL 2>
```

Penjelasan :

Kode ini menghitung **jumlah bilangan kelipatan 4** yang dimasukkan oleh pengguna secara berulang hingga pengguna memasukkan bilangan negatif. Berikut langkah-langkahnya:

1. **Fungsi jumlahKelipatan4:**
 - Meminta pengguna memasukkan angka.
 - Jika angka negatif, proses berhenti dan hasil total dikembalikan.
 - Jika angka positif dan kelipatan 4, angka ditambahkan ke total.
 - Rekursi digunakan untuk melanjutkan proses hingga bilangan negatif dimasukkan.
2. **Input Data** – Pengguna terus memasukkan bilangan satu per satu.
3. **Output** – Menampilkan total jumlah bilangan kelipatan 4 setelah proses selesai.

SOAL TIPE C :

1. Digit

```
package main

import (
    "fmt"
    "strconv"
)

func main() {
    var bilangan_10311240073 int

    fmt.Print("Masukkan bilangan bulat positif (>10): ")
    fmt.Scan(&bilangan_10311240073)

    bilanganStr := strconv.Itoa(bilangan_10311240073)
    panjang := len(bilanganStr)

    tengah := panjang / 2

    var kiriStr, kananStr string
    if panjang%2 == 0 {
        kiriStr = bilanganStr[:tengah]
        kananStr = bilanganStr[tengah:]
    } else {
        kiriStr = bilanganStr[:tengah+1]
        kananStr = bilanganStr[tengah+1:]
    }

    kiri, _ := strconv.Atoi(kiriStr)
    kanan, _ := strconv.Atoi(kananStr)

    hasilPenjumlahan := kiri + kanan
```

```

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

```

```

PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2> go run
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6
PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2> go run
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168

```

Penjelasan :

1. Input Data

- Program meminta pengguna untuk memasukkan **bilangan bulat positif lebih dari 10**. Input ini disimpan dalam variabel `bilangan_103112400073`.

2. Konversi ke String

- Menggunakan `strconv.Itoa`, angka tersebut diubah menjadi bentuk string (`bilanganStr`), yang memungkinkan kita untuk menghitung panjangnya dan membagi angka menjadi dua bagian.

3. Hitung Panjang dan Titik Tengah

- Panjang string dihitung menggunakan `len(bilanganStr)`, dan titik tengah dihitung dengan `panjang / 2`.
- Jika panjang string **genap**, angka dibagi sama rata (misalnya, "1234" menjadi "12" dan "34").
- Jika panjang string **ganjil**, bagian kiri lebih panjang satu digit dibanding bagian kanan (misalnya, "12345" menjadi "123" dan "45").

4. Pemotongan String

- String dibagi menjadi dua bagian menggunakan slicing:
 - `kiriStr`: Bagian awal string.

- `kananStr`: Bagian akhir string.
- Pembagian ini dilakukan dengan mempertimbangkan panjang string (genap atau ganjil).

5. Konversi ke Integer

- Kedua bagian string (`kiriStr` dan `kananStr`) dikonversi kembali ke bilangan integer menggunakan `strconv.Atoi`. Ini penting agar kedua bilangan dapat dijumlahkan.

6. Perhitungan dan Output

- Bagian kiri (`kiri`) dan kanan (`kanan`) dijumlahkan untuk mendapatkan hasil penjumlahan (`hasilPenjumlahan`).
- Program mencetak hasil berupa:
 - Bilangan 1 (bagian kiri).
 - Bilangan 2 (bagian kanan).
 - Hasil penjumlahan kedua bilangan.

2. Bukber IF

```
package main

import "fmt"

func main() {
    var n_103112400073, kode, jumA, jumB, jumC int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&n_103112400073)
    for i := 1; i <= n_103112400073; i++ {
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i)
        fmt.Scan(&kode)

        if bilanganSama(kode) {
            fmt.Println("Hadiah A")
            jumA++
        } else if bilanganBeda(kode) {
            fmt.Println("Hadiah B")
            jumB++
        } else {
            fmt.Println("Hadiah C")
            jumC++
        }
    }
    fmt.Println()
    fmt.Println("Jumlah yang memperoleh Hadiah A:", jumA)
    fmt.Println("Jumlah yang memperoleh Hadiah B:", jumB)
    fmt.Println("Jumlah yang memperoleh Hadiah C:", jumC)
}

func bilanganSama(a int) bool {
    var b1, b2, b3 int
    b1 = a / 100
    b2 = (a / 10) % 10
    b3 = a % 10
    if b1 == b2 && b2 == b3 && b3 == b1 {
        return true
    }
    return false
}

func bilanganBeda(a int) bool {
    var b1, b2, b3 int
    b1 = a / 100
```

```

    b2 = (a / 10) % 10
    b3 = a % 10
    if b1 != b2 && b2 != b3 && b3 != b1 {
        return true
    }
    return false
}

```

```

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
PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2>

```

Penjelasan :

- **Input Jumlah Peserta:** Program meminta input jumlah peserta melalui `fmt.Scan(&n_103112400073)`.
- **Iterasi Peserta:** Dengan menggunakan loop `for`, program meminta nomor kartu dari setiap peserta.
- **Logika Hadiah:**
 - **Fungsi `bilanganSama(kode)`:**
 - Memeriksa apakah semua digit dalam nomor kartu adalah sama (contoh: 111).

- Jika benar, peserta mendapat *Hadiah A* dan variabel `jumA` bertambah 1.
- Fungsi `bilanganBeda(kode)`:
 - Memeriksa apakah semua digit dalam nomor kartu berbeda satu sama lain (contoh: 123).
 - Jika benar, peserta mendapat *Hadiah B* dan variabel `jumB` bertambah 1.
- Jika tidak memenuhi dua kondisi di atas, peserta mendapat *Hadiah C* dan variabel `jumC` bertambah 1.

Fungsi `bilanganSama(a int)` dan `bilanganBeda(a int)`

- Kedua fungsi ini digunakan untuk memeriksa kondisi digit nomor kartu:
 - `bilanganSama`:
 - Nomor kartu dibagi menjadi tiga digit menggunakan operasi matematika:
 - $b1 = a / 100$ (digit ratusan).
 - $b2 = (a / 10) \% 10$ (digit puluhan).
 - $b3 = a \% 10$ (digit satuan).
 - Memastikan semua digit sama: $b1 == b2 \ \&\& \ b2 == b3$.
 - `bilanganBeda`:
 - Proses serupa digunakan untuk membagi nomor menjadi digit.
 - Memastikan semua digit berbeda satu sama lain: $b1 != b2 \ \&\& \ b2 != b3 \ \&\& \ b3 != b1$.

Output

- Program mencetak jumlah peserta yang mendapatkan masing-masing hadiah:
 - Jumlah peserta dengan *Hadiah A* (`jumA`).

- Jumlah peserta dengan *Hadiah B* (jumB).
- Jumlah peserta dengan *Hadiah C* (jumC).

Contoh

Misalnya, jika jumlah peserta adalah 3 dan nomor kartu mereka:

- Peserta ke-1: 111 → Hadiah A.
- Peserta ke-2: 123 → Hadiah B.
- Peserta ke-3: 122 → Hadiah C.

Output akan menjadi:

Jumlah yang memperoleh Hadiah A: 1

Jumlah yang memperoleh Hadiah B: 1

Jumlah yang memperoleh Hadiah C: 1

3. Perkalian Dengan Cara Penjumlahan

```
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_10311240073, m int

    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&n_10311240073)
    fmt.Print("Masukkan bilangan m: ")
    fmt.Scan(&m)

    negatif := false
    if m < 0 {
        m = -m
        negatif = true
    }

    hasil := perkalian(n_10311240073, m, 0)

    if negatif {
        hasil = -hasil
    }

    fmt.Printf("Hasil dari %d x %d = %d\n", n_10311240073, m,
hasil)
}
```

```
dengan cara Penjumlahan.go"
```

```
Masukkan bilangan n: 5
```

```
Masukkan bilangan m: 6
```

```
Hasil dari 5 x 6 = 30
```

```
PS D:\ALPRO 2\103112400073_LATIHAN SOAL 2> █
```

Penjelasan :

Kode ini menghitung hasil perkalian dua bilangan secara **rekursif**, dengan langkah-langkah berikut:

1. **Fungsi Rekursif perkalian:**
 - Fungsi ini melakukan perkalian bilangan n dengan m tanpa menggunakan operator $*$.
 - Jika m sudah bernilai 0, penghitungan selesai dan mengembalikan hasil akhir.
 - Jika tidak, fungsi menambahkan n ke $hasil$ dan memanggil dirinya sendiri dengan nilai m yang berkurang satu.
2. **Penanganan Bilangan Negatif:**
 - Jika bilangan kedua (m) negatif, program mengubahnya menjadi positif dan menyimpan status negatif dalam variabel `negatif`.
 - Setelah penghitungan selesai, hasil dikembalikan sebagai bilangan negatif jika diperlukan.
3. **Input Data:**
 - Pengguna memasukkan dua bilangan: n (bilangan pertama) dan m (bilangan kedua).
4. **Output:**
 - Program mencetak hasil perkalian dalam format $n \times m = hasil$.

Contoh Kasus:

- Jika pengguna memasukkan $n = 5$ dan $m = -3$, program akan:
 - Mengubah m menjadi positif: 3.
 - Menghitung hasil: $5 + 5 + 5 = 15$.
 - Karena m negatif, hasil akhir menjadi -15.
 - Output: Hasil dari $5 \times -3 = -15$.