

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
LATIHAN SOAL**



Oleh:

NAMA: ICHYA ULUMIDDIIN

NIM: 103112400076

KELAS: 12IF-01-A

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. TIPE A

Source Code No. 1

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jam, menit int
    var isMember bool
    var voucher string
    var NIM string = "103112400076"
    fmt.Println("NIM :", NIM)
    fmt.Print("Durasi (jam) : ")
    fmt.Scanln(&jam)
    fmt.Print("Durasi (menit) : ")
    fmt.Scanln(&menit)
    fmt.Print("Apakah Member? (true/false) : ")
    fmt.Scanln(&isMember)
    fmt.Print("Masukkan no voucher (jika ada) : ")
    fmt.Scanln(&voucher)
    totalJam := hitungDurasi(jam, menit)
    tarif := 5000
    if isMember {
        tarif = 3500
    }
    totalBiaya := float64(tarif * totalJam)
    if totalJam >= 3 && adaDiskon(voucher) {
        diskon := totalBiaya * 0.166666667 // diskon 16.666...%
        totalBiaya -= diskon
    }
    totalBiaya = math.Round(totalBiaya*100) / 100
    if totalJam >= 3 && adaDiskon(voucher) {
        fmt.Printf("Biaya sewa setelah diskon (jika memenuhi syarat) : Rp %.2f\n", totalBiaya)
    } else {
        fmt.Printf("Biaya sewa : Rp %.2f\n", totalBiaya)
    }
}

func hitungDurasi(jam, menit int) int {
    if jam == 0 {
        if menit < 10 {
            return 0
        }
        return 1
    }
    if menit >= 10 {
        return jam + 1
    }
    return jam
}

func adaDiskon(voucher string) bool {
    for _, char := range voucher {
        if char == '5' || char == '6' {
            return true
        }
    }
    return false
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\Alpro 2> go run "c
NIM : 103112400076
Durasi (jam) : 2
Durasi (menit) : 30
Apakah Member? (true/false) : true
Masukkan no voucher (jika ada) : 123456
Biaya sewa setelah diskon (jika memenuhi syarat) : Rp 8750.00
```

Penjelasan

Program Go ini menghitung biaya sewa berdasarkan durasi penggunaan, status member, dan penggunaan voucher. Pengguna memasukkan durasi (jam dan menit), status member, dan kode voucher. Tarif sewa berbeda untuk member (Rp 3500/jam) dan non-member (Rp 5000/jam). Jika durasi ≥ 3 jam dan voucher mengandung angka 5 atau 6, maka diskon 16,67% diberikan. Program juga mencetak NIM (`103112400076`) sebagai identitas. Hasil akhir berupa total biaya sewa, dengan atau tanpa diskon.

Source Code No. 2

```
package main

import (
    "fmt"
)

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

func main() {
    var a, b int
    var NIM string = "103112400076"
    fmt.Println("NIM :", NIM)
    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)
    fmt.Printf("Perfect numbers antara %d dan %d: ", a,
b)

    ditemukan := false
    for i := a; i <= b; i++ {
        if isPerfectNumber(i) {
            fmt.Printf("%d ", i)
            ditemukan = true
        }
    }
    if !ditemukan {
        fmt.Print("Tidak ada")
    }
    fmt.Println()
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\A.  
NIM : 103112400076  
Masukkan nilai a: 3  
Masukkan nilai b: 13  
Perfect numbers antara 3 dan 13: 6
```

Penjelasan

Program Go ini digunakan untuk mencari dan menampilkan bilangan perfect number dalam rentang `a` hingga `b`. Di awal, program menampilkan NIM (`103112400076`) sebagai identitas. Pengguna diminta memasukkan dua bilangan bulat positif, yaitu `a` dan `b`.

Program menggunakan fungsi `isPerfectNumber(n int)` untuk memeriksa apakah suatu bilangan `n` adalah perfect number, yaitu jika jumlah semua pembaginya (selain dirinya sendiri) sama dengan `n`. Selanjutnya, program melakukan iterasi dari `a` hingga `b`, dan mencetak semua bilangan yang memenuhi syarat tersebut. Jika tidak ditemukan satupun, program akan mencetak "Tidak ada".

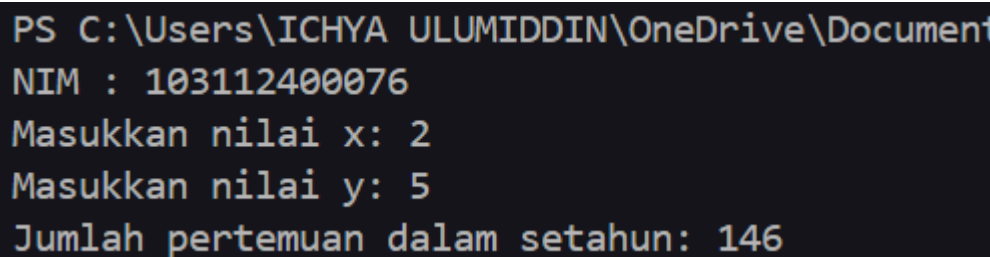
Source Code No. 3

```
package main

import (
    "fmt"
)

func main() {
    var x, y int
    var NIM string = "103112400076"
    fmt.Println("NIM :", NIM)
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y)
    jumlahPertemuan := 0
    for hari := 1; hari <= 365; hari++ {
        if hari%x == 0 && hari%y != 0 {
            jumlahPertemuan++
        }
    }
    fmt.Printf("Jumlah pertemuan dalam setahun: %d\n",
        jumlahPertemuan)
}
```

Output



```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Document
NIM : 103112400076
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan dalam setahun: 146
```

Penjelasan

Program Go di atas digunakan untuk menghitung jumlah hari pertemuan dalam satu tahun (365 hari) berdasarkan dua bilangan bulat `x` dan `y` yang dimasukkan oleh pengguna. Di awal program, ditampilkan NIM `103112400076` sebagai identitas. Nilai `x` menyatakan hari beberapa pertemuan diadakan (kelipatan `x`), sementara `y` merupakan hari-hari yang tidak boleh digunakan untuk pertemuan (kelipatan `y`).

Program kemudian melakukan perulangan dari hari ke-1 hingga hari ke-365, dan menghitung jumlah hari yang merupakan kelipatan dari x namun bukan kelipatan dari y . Setiap hari yang memenuhi syarat tersebut akan dihitung sebagai hari pertemuan. Hasil akhirnya ditampilkan sebagai “Jumlah pertemuan dalam setahun”.

II. TIPE B

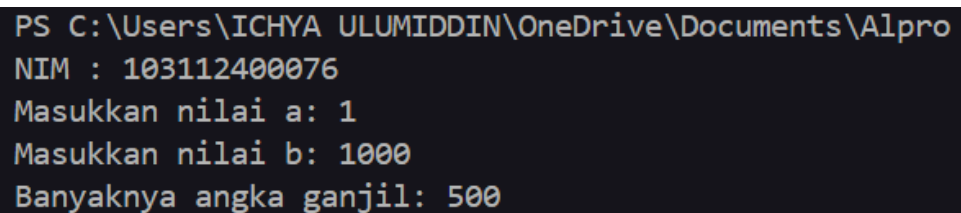
Source Code No. 1

```
package main

import (
    "fmt"
)

func main() {
    var a, b int
    var NIM string = "103112400076"
    fmt.Println("NIM :", NIM)
    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\ICHYA ULUMIDDIN\OneDrive\Documents\Alpro
NIM : 103112400076
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500
```

Penjelasan

Program Go di atas bertujuan untuk menghitung jumlah bilangan ganjil dalam sebuah rentang dari `a` hingga `b`, di mana kedua nilai tersebut dimasukkan oleh pengguna. Program diawali dengan menampilkan NIM

`103112400076` sebagai identitas. Setelah itu, pengguna diminta untuk memasukkan dua bilangan bulat `a` dan `b`.

Program kemudian melakukan perulangan dari nilai `a` sampai `b`, dan memeriksa setiap bilangan menggunakan kondisi $i \% 2 \neq 0$ untuk menentukan apakah bilangan tersebut ganjil. Jika iya, maka penghitung `jumlahGanjil` akan ditambah satu. Setelah seluruh bilangan dalam rentang diperiksa, hasil jumlah bilangan ganjil akan ditampilkan ke layar.

Source Code No. 2

```
package main

import (
    "fmt"
)

func main() {
    var m int
    var NIM string = "103112400076"
    fmt.Println("NIM :", NIM)
    fmt.Print("Masukkan jumlah rombongan: ")
    fmt.Scan(&m)
    for i := 1; i <= m; i++ {
        var menu, orang, sisa int
        fmt.Printf("Masukkan jumlah menu, jumlah
orang, dan status sisa makanan (0 untuk tidak, 1 untuk
iya):\n: ")
        fmt.Scan(&menu, &orang, &sisa)
        tarif := 0
        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\ICHYA ULUMIDDIN\OneDrive\Documents\Alpro 2> go run "c:\Users\ICHYA ULUMIDDIN\On
NIM : 103112400076
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
```

Penjelasan

Program Go di atas digunakan untuk menghitung total biaya makan untuk beberapa rombongan dalam suatu acara, berdasarkan jumlah menu yang dipesan, jumlah orang, dan apakah ada sisa makanan. Program dimulai dengan menampilkan NIM `103112400076` sebagai identitas pengguna. Selanjutnya, pengguna diminta memasukkan jumlah rombongan (`m`) yang akan diproses.

Untuk setiap rombongan, pengguna menginput jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak ada sisa, 1 untuk ada sisa). Biaya dasar ditentukan berdasarkan jumlah menu: jika menu ≤ 3 , biaya tetap Rp 10.000; jika menu antara 4 hingga 50, maka ditambahkan Rp 2.500 untuk setiap menu di atas 3; dan jika lebih dari 50 menu, tarif tetap Rp 100.000. Apabila makanan tidak habis (sisa = 1), maka total tarif dikalikan dengan jumlah orang sebagai bentuk penalti.

Source Code No. 3

```
package main

import (
    "fmt"
)

func inputDanHitung(total int) int {
    var angka int
    fmt.Scan(&angka)
    if angka < 0 {
        return total
    }
    fmt.Printf("%d ", angka)
    if angka%4 == 0 {
        total += angka
    }
    return inputDanHitung(total)
}

func main() {
    NIM := "103112400076"
    fmt.Println("NIM:", NIM)
    fmt.Print("Masukkan bilangan (negatif untuk
berhenti):\n: ")
    total := inputDanHitung(0)
    fmt.Println()
    fmt.Printf("Jumlah bilangan kelipatan 4: %d\n",
total)
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\
NIM: 103112400076
Masukkan bilangan (negatif untuk berhenti):
: 2 3 4 5 6 -1
2 3 4 5 6
Jumlah bilangan kelipatan 4: 4
```

Penjelasan

Program Go di atas digunakan untuk menghitung jumlah dari bilangan-bilangan positif kelipatan 4 yang dimasukkan oleh pengguna secara berulang. Di awal program, dicetak NIM `103112400076` sebagai identitas. Pengguna diminta memasukkan bilangan satu per satu melalui terminal, dan proses input akan terus berulang sampai pengguna memasukkan bilangan negatif, yang menjadi tanda untuk berhenti. Fungsi `inputDanHitung` merupakan fungsi rekursif tail-end yang membaca satu angka, mencetak angka tersebut secara menyamping, lalu memeriksa apakah angka tersebut merupakan kelipatan 4. Jika ya, maka angka ditambahkan ke variabel `total`. Fungsi akan terus memanggil dirinya sendiri sampai ditemukan input negatif.

Tujuan dari program ini adalah untuk menjumlahkan semua bilangan positif kelipatan 4 dari serangkaian input yang dimasukkan pengguna, sambil menampilkan semua angka yang telah dimasukkan dalam satu baris.

III. TIPE C

Source Code No. 1

```
package main

import (
    "fmt"
    "math"
)

func main() {
    NIM := "103112400076"
    fmt.Println("NIM:", NIM)
    var input int
    fmt.Print("Masukkan bilangan bulat positif (>10): ")
    fmt.Scan(&input)
    jumlahDigit := 0
    temp := input
    for temp > 0 {
        jumlahDigit++
        temp /= 10
    }
    var kananDigit int
    if jumlahDigit%2 == 0 {
        kananDigit = jumlahDigit / 2
    } else {
        kananDigit = jumlahDigit / 2
    }
    pangkat := int(math.Pow(10,
float64(kananDigit)))
    kiri := input / pangkat
    kanan := input % pangkat
    fmt.Println("Bilangan 1:", kiri)
    fmt.Println("Bilangan 2:", kanan)
    fmt.Println("Hasil penjumlahan:", kiri+kanan)
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\Alpro 21  
NIM: 103112400076  
Masukkan bilangan bulat positif (>10): 12345  
Bilangan 1: 123  
Bilangan 2: 45  
Hasil penjumlahan: 168
```

Penjelasan

Program Go di atas bertujuan untuk memisahkan sebuah bilangan bulat positif menjadi dua bagian berdasarkan jumlah digitnya, lalu menjumlahkan kedua bagian tersebut. Program dimulai dengan mencetak NIM `103112400076` sebagai identitas. Pengguna diminta memasukkan sebuah bilangan bulat positif yang memiliki lebih dari dua digit.

Langkah pertama, program menghitung jumlah digit dari bilangan tersebut menggunakan perulangan pembagian. Setelah jumlah digit diketahui, program menentukan jumlah digit bagian kanan (`kananDigit`), yaitu setengah dari total digit (dibulatkan ke bawah jika jumlah digit ganjil). Bagian kiri otomatis menjadi sisanya. Untuk memisahkan kedua bagian, digunakan operasi matematika: bagian kiri diperoleh dengan membagi bilangan dengan $10^{\text{kananDigit}}$, sedangkan bagian kanan diperoleh dengan mengambil sisa pembagian (`mod`) dari $10^{\text{kananDigit}}$.

Source Code No. 2

```
package main
import (
    "fmt"
)
func semuaSama(n int) bool {
    digit := n % 10
    n /= 10
    for n > 0 {
        if n%10 != digit {
            return false
        }
        n /= 10
    }
    return true
}
func semuaBerbeda(n int) bool {
    angka := 0
    for n > 0 {
        d := n % 10
        if angka & (1 << d) != 0 {
            return false
        }
        angka |= 1 << d
        n /= 10
    }
    return true
}
func main() {
    NIM := "10311240076"
    fmt.Println("NIM:", NIM)
    var n int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&n)
    hadiahA := 0
    hadiahB := 0
    hadiahC := 0
    for i := 1; i <= n; i++ {
        var nomor int
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i)
        fmt.Scan(&nomor)
        if semuaSama(nomor) {
            fmt.Println("Hadiah A")
            hadiahA++
        } else if semuaBerbeda(nomor) {
            fmt.Println("Hadiah B")
            hadiahB++
        } else {
            fmt.Println("Hadiah C")
            hadiahC++
        }
    }
    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)
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents
NIM: 103112400076
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
```

Penjelasan

Program Go di atas digunakan untuk mengelompokkan peserta berdasarkan pola angka pada nomor kartu yang mereka miliki. Program diawali dengan menampilkan NIM `103112400076` sebagai identitas. Selanjutnya, pengguna diminta memasukkan jumlah peserta (`n`). Untuk setiap peserta, pengguna akan menginput nomor kartu peserta, yang akan diproses untuk menentukan jenis hadiah yang diperoleh.

Program menggunakan dua fungsi: `semuaSama(n int)` untuk memeriksa apakah semua digit dalam nomor kartu adalah sama, dan `semuaBerbeda(n int)` untuk memeriksa apakah semua digit berbeda satu sama lain. Jika semua digit sama, peserta mendapat Hadiah A; jika semua digit berbeda, peserta mendapat Hadiah B; dan jika tidak termasuk keduanya (ada digit yang sama dan berbeda), maka peserta mendapat Hadiah C.

Source Code No. 3

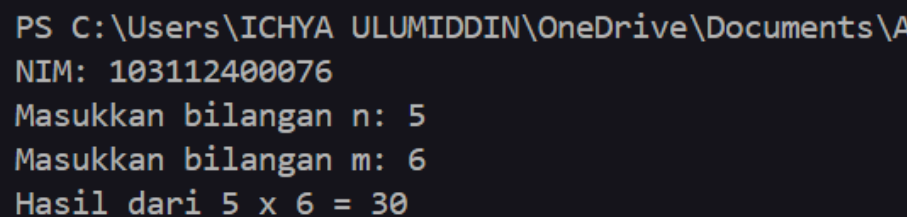
```
package main

import (
    "fmt"
)

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

func main() {
    NIM := "103112400076"
    fmt.Println("NIM:", NIM)
    var n, m int
    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&n)
    fmt.Print("Masukkan bilangan m: ")
    fmt.Scan(&m)
    hasil := kali(n, m, 0)
    fmt.Printf("Hasil dari %d x %d = %d\n", n, m,
    hasil)
}
```

Output



```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\A
NIM: 103112400076
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Hasil dari 5 x 6 = 30
```

Penjelasan

Program Go di atas digunakan untuk menghitung hasil perkalian dua bilangan bulat positif `n` dan `m` menggunakan rekursi tail-end yang berbasis penjumlahan berulang. Program dimulai dengan menampilkan NIM `103112400076` sebagai identitas pengguna. Pengguna kemudian diminta untuk memasukkan dua bilangan `n` dan `m`.

Fungsi `kali(n, m, hasil)` bekerja secara rekursif untuk menambahkan `n` sebanyak `m` kali. Setiap pemanggilan fungsi akan mengurangi nilai `m` satu per satu sambil menambahkan `n` ke parameter `hasil`, hingga `m` bernilai nol, di mana hasil akhir dikembalikan. Ini mencerminkan prinsip dasar perkalian sebagai penjumlahan berulang.