

ALGORITMA DAN PEMROGRAMAN 2

LATIHAN SOAL 2



Oleh:

MUHAMMAD GAMEL AL GHIFARI

103112400028

12IF-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

1. Konsep Dasar Pemrograman dengan Go

Go (Golang) adalah bahasa pemrograman yang dikembangkan oleh Google dengan fokus pada kesederhanaan, efisiensi, dan konkurensi. Beberapa fitur utamanya meliputi:

Static typing: Tipe data ditentukan saat kompilasi.

Garbage collection: Manajemen memori otomatis.

Funsgi multiple return: Kemampuan mengembalikan lebih dari satu nilai.

Package system: Modularitas melalui impor paket seperti fmt untuk I/O.

2. Algoritma dan Struktur Data

Program-program dalam laporan menggunakan konsep:

Looping (for): Untuk iterasi rentang bilangan atau input.

Rekursi: Contoh pada perkalian dan penjumlahan bilangan.

Kondisional (if, switch): Penyeleksian kondisi seperti diskon atau penentuan hadiah.

Array/Slice: Penyimpanan data kelompok (nomor peserta, biaya rombongan).

3. Operasi Aritmatika dan Logika

Pembagian bilangan: Program memotong bilangan menjadi dua bagian dengan operasi / (divisi) dan % (modulo).

Perfect numbers: Bilangan yang jumlah faktornya sama dengan bilangan itu sendiri (contoh: $6 = 1+2+3$).

Perkalian rekursif: Mengganti operasi * dengan penjumlahan berulang ($a * b = a + a*(b-1)$).

4. Validasi Input

Filter bilangan positif: Memastikan input >10 untuk program pemotongan bilangan.

Terminasi negatif: Program keluar jika input bilangan negatif (pada penjumlahan kelipatan 4).

5. Contoh Aplikasi

Sistem sewa sepeda: Menggabungkan durasi, membership, dan diskon.

Sistem hadiah: Analisis pola digit untuk klasifikasi (contoh: semua digit sama = hadiah A).

6. Kompleksitas Waktu

$O(n)$: Pada pencarian bilangan ganjil atau perfect numbers.

$O(\log n)$: Pemotongan bilangan berdasarkan digit.

$O(1)$: Operasi aritmatika sederhana seperti perkalian.

II. UNGUIDED

TYPE A:

1.

```
// MUAHAMMAD GAMEL AL GHIFARI

// 103112400028

package main

import (
    "fmt"
)

func main() {
    var jam, menit int
    var member bool
    var voucher 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(&member)

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

```

totalJam := hitungTotalJam(jam, menit)

biaya := hitungBiaya(totalJam, member)

if totalJam > 3 {
    biaya = biaya * 0.9
}

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

func hitungTotalJam(jam, menit int) float64 {
    totalJam := float64(jam)
    if menit >= 10 || totalJam < 1 {
        totalJam += float64(menit) / 60.0
    }
    return totalJam
}

func hitungBiaya(totalJam float64, member bool) float64 {
    tarif := 5000.0
    if member {
        tarif = 3500.0
    }
}

```

```
        return totalJam * tarif
    }
```

OUTPUT:

```
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\LATIHANSOAL2\1a.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 8750.00
```

DESKRIPSI:

Program ini menghitung biaya sewa sepeda berdasarkan:

Durasi: Jam dan menit (menit ≥ 10 dihitung sebagai pecahan jam)

Membership:

Member: Rp3.500/jam

Non-member: Rp5.000/jam

Diskon:

Diskon 10% jika sewa > 3 jam

2.

```
// MUHAMMAD GAMEL AL GHIFARI
// 103112400028

package main

import "fmt"
```

```
func main() {  
    var start, end int  
  
    fmt.Print("Masukkan nilai a: ")  
  
    fmt.Scan(&start)  
  
    fmt.Print("Masukkan nilai b: ")  
  
    fmt.Scan(&end)  
  
  
    fmt.Printf("Perfect numbers antara %d dan %d: ", start, end)  
  
  
    hasPerfect := printPerfectNumbers(start, end)  
  
  
    if !hasPerfect {  
        fmt.Print("Tidak ada")  
    }  
  
    fmt.Println()  
}  
  
func printPerfectNumbers(start, end int) bool {  
    found := false  
  
    for n := start; n <= end; n++ {  
        if checkPerfect(n) {  
            fmt.Printf("%d ", n)  
  
            found = true  
        }  
    }  
}
```

```

    }

    return found
}

func checkPerfect(n int) bool {

    if n < 2 {

        return false

    }

    total := 0

    for d := 1; d <= n/2; d++ {

        if n%d == 0 {

            total += d

        }

    }

    return total == n

}

```

OUTPUT:

```

PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\LATIHANSOAL2\1b.go"
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6

```

DESKRIPSI:

Program ini mencari bilangan sempurna (perfect numbers) dalam rentang antara dua angka yang dimasukkan pengguna. Bilangan sempurna adalah bilangan yang jumlah faktor pembaginya (tidak termasuk bilangan itu sendiri) sama dengan bilangan tersebut.

3.


```
// MUHAMMAD GAMEL AL GHIFARI

// 103112400028

package main

import "fmt"

func main() {

    var intervalA, intervalB int

    fmt.Print("Masukkan nilai x: ")

    fmt.Scan(&intervalA)

    fmt.Print("Masukkan nilai y: ")

    fmt.Scan(&intervalB)

    totalMeetings := calculateSecretMeetings(intervalA, intervalB)

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

func calculateSecretMeetings(a, b int) int {

    meetingCount := 0

    for day := 1; day <= 365; day++ {

        if meetsCriteria(day, a, b) {

            meetingCount++

        }

    }

}
```

```

    }

    return meetingCount
}

func meetsCriteria(day, x, y int) bool {
    return day%x == 0 && day%y != 0
}

```

OUTPUT:

```

PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\LATIHANSOAL2\1c.go"
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan rahasia dalam setahun: 146

```

DESKRIPSI:

Program ini menghitung jumlah pertemuan rahasia yang terjadi dalam setahun (365 hari) berdasarkan dua input pengguna, yaitu x dan y

TYPE B

1.

```

// MUHAMMAD GAMEL AL GHIFARI

// 103112400028

package main

import "fmt"

func countOddNumbers(lowerBound, upperBound int) int {
    oddCount := 0

```

```

        for number := lowerBound; number <= upperBound; number++ {
            if isOdd(number) {
                oddCount++
            }
        }
        return oddCount
    }

func isOdd(n int) bool {
    return n%2 != 0
}

func main() {
    upperLimit := 1000
    oddNumbersCount := countOddNumbers(1, upperLimit)
    fmt.Printf("Total bilangan ganjil antara 1 dan %d: %d\n", upperLimit,
    oddNumbersCount)
}

```

OUTPUT:

```

PS D:\LAPRAK\> go run "d:\LAPRAK\LATIHANSOAL2\tempCodeRunnerFile.go"
Total bilangan ganjil antara 1 dan 1000: 500

```

DESKRIPSI:

Program ini menghitung banyaknya bilangan ganjil dalam rentang antara dua angka yang dimasukkan pengguna (a dan b).

2.

```
// MUHAMMAD GAMEL AL GHIFARI

// 103112400028

package main

import "fmt"

const (
    basePrice    = 10000
    additionalPrice = 2500
    bulkPrice     = 100000
    baseThreshold = 3
    bulkThreshold = 50
)

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

    groupCosts := calculateAllGroupCosts(groupCount)
    printResults(groupCosts)
}

func calculateAllGroupCosts(groupCount int) []int {
```

```
costs := make([]int, groupCount)

for i := 0; i < groupCount; i++ {

    fmt.Printf("\nData rombongan ke-%d:\n", i+1)

    menuCount, peopleCount, hasLeftover := getGroupInput()

    costs[i] = calculateGroupCost(menuCount, peopleCount,
hasLeftover)

}

return costs

}

func getGroupInput() (int, int, bool) {

    var menuCount, peopleCount int

    var hasLeftover bool

    fmt.Print("Jumlah menu: ")

    fmt.Scan(&menuCount)

    fmt.Print("Jumlah orang: ")

    fmt.Scan(&peopleCount)

    fmt.Print("Ada sisa? (true/false): ")

    fmt.Scan(&hasLeftover)

    return menuCount, peopleCount, hasLeftover

}
```

```
func calculateGroupCost(menuCount, peopleCount int, hasLeftover bool) int {  
    var totalCost int  
  
    switch {  
    case menuCount > bulkThreshold:  
        totalCost = bulkPrice  
    case menuCount > baseThreshold:  
        totalCost = basePrice + (menuCount-  
baseThreshold)*additionalPrice  
    default:  
        totalCost = basePrice  
    }  
  
    if hasLeftover {  
        totalCost *= peopleCount  
    }  
  
    return totalCost  
}  
  
func printResults(costs []int) {  
    fmt.Println("\nHasil Perhitungan Biaya:")  
    for i, cost := range costs {  
        fmt.Printf("Rombongan ke-%d: Rp %d\n", i+1, cost)  
    }  
}
```

```
}
```

OUTPUT:

```
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\LATIHANSOAL2\tempCodeRunnerFile.go"
Masukkan jumlah rombongan: 2

Data rombongan ke-1:
Jumlah menu: 3
Jumlah orang: 12
Ada sisa? (true/false): 0

Data rombongan ke-2:
Jumlah menu: 2
Jumlah orang: 15
Ada sisa? (true/false): 1

Hasil Perhitungan Biaya:
Rombongan ke-1: Rp 10000
Rombongan ke-2: Rp 150000
```

DESKRIPSI:

Program ini menghitung biaya yang harus dibayar oleh beberapa rombongan berdasarkan jumlah menu yang dipesan, jumlah orang, dan apakah ada sisa makanan.

3.

```
// MUHAMMAD GAMEL AL GHIFARI

// 103112400028

package main

import "fmt"

func main() {
    printProgramHeader()
    total := calculateMultiplesOfFour()
    printResult(total)
```

```
}

func printProgramHeader() {
    fmt.Println("Program menjumlahkan bilangan kelipatan 4")
    fmt.Println("Masukkan bilangan (akhiri dengan bilangan negatif):")
}

func calculateMultiplesOfFour() int {
    return sumMultiplesRecursively(0)
}

func sumMultiplesRecursively(accumulator int) int {
    number := readInputNumber()

    if isTerminationCondition(number) {
        return accumulator
    }

    if isMultipleOfFour(number) {
        return sumMultiplesRecursively(accumulator + number)
    }

    return sumMultiplesRecursively(accumulator)
}
```



```

func readInputNumber() int {
    var number int
    fmt.Scan(&number)
    return number
}

func isTerminationCondition(number int) bool {
    return number < 0
}

func isMultipleOfFour(number int) bool {
    return number > 0 && number%4 == 0
}

func printResult(total int) {
    fmt.Printf("Total jumlah bilangan kelipatan 4: %d\n", total)
}

```

OUTPUT:

```

PS D:\LAPRAKKKK> go run "d:\LAPRAKKKK\LATIHANSOAL2\2c.go"
Program menjumlahkan bilangan kelipatan 4
Masukkan bilangan (akhiri dengan bilangan negatif):
2 3 4 5 6 -1
Total jumlah bilangan kelipatan 4: 4

```

DESKRIPSI:

Program ini menghitung total bilangan kelipatan 4 yang dimasukkan pengguna secara rekursif.

TYPE C

1.

```
// MUHAMMAD GAMEL AL GHIFARI
// 103112400028

package main

import "fmt"

func main() {
    bilangan := mintaInputBilangan()
    bagian1, bagian2 := potongBilangan(bilangan)
    hasil := hitungJumlah(bagian1, bagian2)
    tampilkanHasil(bagian1, bagian2, hasil)
}

func mintaInputBilangan() int {
    var n int
    for {
        fmt.Print("Masukkan bilangan bulat positif (>10): ")
        fmt.Scan(&n)
        if n > 10 {
            return n
        }
    }
}
```

```
        fmt.Println("Error: Bilangan harus >10")
    }
}

func potongBilangan(n int) (int, int) {
    // Hitung jumlah digit
    digit := 0
    temp := n
    for temp > 0 {
        temp /= 10
        digit++
    }

    // Hitung titik potong
    titik := digit / 2
    if digit%2 != 0 {
        titik++
    }

    // Pisahkan bilangan
    pembagi := 1
    for i := 0; i < digit-titik; i++ {
        pembagi *= 10
    }
}
```

```

        return n / pembagi, n % pembagi
    }

func hitungJumlah(b1, b2 int) int {
    return b1 + b2
}

func tampilkanHasil(b1, b2, hasil int) {
    fmt.Printf("Bilangan 1: %d\n", b1)
    fmt.Printf("Bilangan 2: %d\n", b2)
    fmt.Printf("Hasil penjumlahan: %d\n", hasil)
}

```

OUTPUT:

```

PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\LATIHANSOAL2\3a.go"
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6

```

DESKRIPSI:

Program ini memproses sebuah bilangan bulat positif dengan cara membaginya menjadi dua bagian dan menjumlahkannya.

2.

```

// MUHAMMAD GAMEL AL GHIFARI

// 103112400028

package main

```

```
import "fmt"

func main() {
    peserta := inputDataPeserta()
    hadiahA, hadiahB, hadiahC := hitungHadiah(peserta)
    tampilkanRingkasan(hadiahA, hadiahB, hadiahC)
}

func inputDataPeserta() []string {
    var jumlahPeserta int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&jumlahPeserta)

    peserta := make([]string, jumlahPeserta)
    for i := 0; i < jumlahPeserta; i++ {
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i+1)
        fmt.Scan(&peserta[i])
    }
    return peserta
}

func hitungHadiah(peserta []string) (int, int, int) {
    var a, b, c int
```

```
for _, nomor := range peserta {  
    hadiah := tentukanJenisHadiah(nomor)  
    fmt.Println("Hadiah", hadiah)  
  
    switch hadiah {  
    case "A":  
        a++  
    case "B":  
        b++  
    case "C":  
        c++  
    }  
}  
return a, b, c  
}
```

```
func tentukanJenisHadiah(nomor string) string {  
    if semuaDigitSama(nomor) {  
        return "A"  
    }  
    if semuaDigitBeda(nomor) {  
        return "B"  
    }  
    return "C"  
}
```

```
}
```

```
func semuaDigitSama(nomor string) bool {
```

```
    for i := 1; i < len(nomor); i++ {
```

```
        if nomor[i] != nomor[0] {
```

```
            return false
```

```
        }
```

```
    }
```

```
    return true
```

```
}
```

```
func semuaDigitBeda(nomor string) bool {
```

```
    seen := make(map[byte]bool)
```

```
    for i := 0; i < len(nomor); i++ {
```

```
        if seen[nomor[i]] {
```

```
            return false
```

```
        }
```

```
        seen[nomor[i]] = true
```

```
    }
```

```
    return true
```

```
}
```

```
func tampilkanRingkasan(a, b, c int) {
```

```
    fmt.Println("\nJumlah yang memperoleh Hadiah A:", a)
```

```
        fmt.Println("Jumlah yang memperoleh Hadiah B:", b)

        fmt.Println("Jumlah yang memperoleh Hadiah C:", c)

    }
```

OUTPUT:

```
PS D:\LAPRAK\> go run "d:\LAPRAK\LATIHANSOAL2\tempCodeRunnerFile.go"
Masukkan jumlah peserta: 3
Masukkan nomor kartu peserta ke-1: 333
Masukkan nomor kartu peserta ke-2: 123
Masukkan nomor kartu peserta ke-3: 898
Hadiah A
Hadiah B
Hadiah C

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

DESKRIPSI:

Program ini menentukan hadiah untuk peserta berdasarkan nomor kartu unik mereka.

3.

```
// MUHAMMAD GAMEL AL GHIFARI

// 103112400028

package main

import "fmt"

func main() {

    bilangan1, bilangan2 := bacaInput()

    hasil := hitungPerkalian(bilangan1, bilangan2)

    tampilkanHasil(bilangan1, bilangan2, hasil)
```



```
}
```

```
func bacaInput() (int, int) {
```

```
    var a, b int
```

```
    fmt.Print("Masukkan bilangan n: ")
```

```
    fmt.Scan(&a)
```

```
    fmt.Print("Masukkan bilangan m: ")
```

```
    fmt.Scan(&b)
```

```
    return a, b
```

```
}
```

```
func hitungPerkalian(a, b int) int {
```

```
    switch {
```

```
        case a == 0 || b == 0:
```

```
            return 0
```

```
        case a == 1:
```

```
            return b
```

```
        case b == 1:
```

```
            return a
```

```
        case a < 0:
```

```
            return -hitungPerkalian(-a, b)
```

```
        default:
```

```
            return b + hitungPerkalian(a-1, b)
```

```
    }
```

```
}  
  
func tampilkanHasil(a, b, hasil int) {  
    fmt.Printf("Hasil dari %d x %d = %d\n", a, b, hasil)  
}
```

OUTPUT:

```
PS D:\LAPRAKKKK> go run "d:\LAPRAKKKK\LATIHANSOAL2\3c.go"  
Masukkan bilangan n: 5  
Masukkan bilangan m: 6  
Hasil dari 5 x 6 = 30
```

DESKRIPSI:

Program ini menghitung hasil perkalian dua bilangan bulat (n dan m) menggunakan pendekatan rekursif.

III. KESIMPULAN

Koleksi program dalam dokumen ini menunjukkan fleksibilitas bahasa Go dalam menyelesaikan beragam tantangan pemrograman. Beberapa poin kunci yang menonjol:

1. **Efisiensi Algoritma:** Penggunaan rekursi (perkalian) dan optimasi perhitungan (bilangan ganjil) meminimalkan kompleksitas.
2. **Kreativitas Solusi:** Seperti pembagian bilangan menjadi dua bagian atau sistem hadiah berbasis pola digit, yang bisa diaplikasikan di sistem undian atau event.
3. **Robustness:** Validasi input dan penanganan kasus tepi (bilangan negatif, input invalid) membuat program lebih tangguh.
4. **Edukatif:** Cocok untuk mahasiswa mempelajari konsep dasar hingga menengah seperti loop, kondisional, dan manajemen memori.

Dokumen ini bukan sekadar kumpulan kode, tetapi contoh nyata bagaimana Go bisa digunakan untuk menyederhanakan solusi komputasional sehari-hari.

IV. REFERENSI

Programmer Zaman Now. "ALGORITMA & STRUKTUR DATA - SEMESTER 2." YouTube, 2023, https://youtu.be/IO_vkyJnMas?si=4jSJTZ5Zd2Fi5Vd2.