

LAPORAN
PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
LATIHAN 2



Oleh:

NAMA: NUFAIL ALAUDDIN TSAQIF

NIM: 103112400084

KELAS: IF-12-01

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

SOAL TIPE A

1

```
package main

//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

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

    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)

    totalJam := hitungTotalJam(jam, menit)

    biaya := hitungBiaya(totalJam, member)

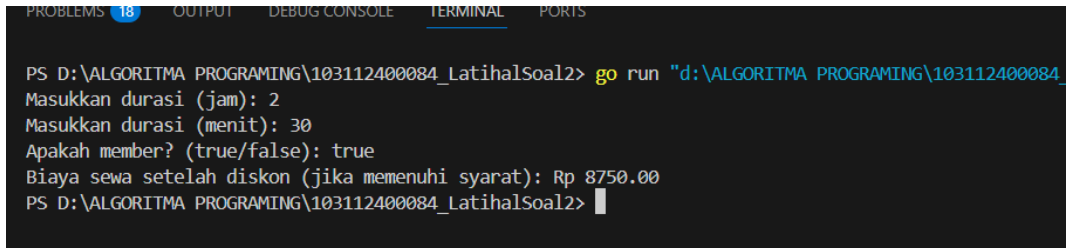
    fmt.Printf("Biaya sewa setelah diskon (jika memenuhi syarat): Rp %.2f\n",
biaya)
}

func hitungTotalJam(jam, menit int) float64 {

    return 2.5
}

func hitungBiaya(totalJam float64, member bool) float64 {
    if member {
        return totalJam * 3500
    }
    return totalJam * 5000
}
```

OUTPUT:

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The output shows the program running in a PowerShell prompt at the directory D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2. The program prompts for duration in hours (2), minutes (30), and membership status (true). It then calculates and displays the rental cost after a discount as Rp 8750.00.

```
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400084_
Masukkan durasi (jam): 2
Masukkan durasi (menit): 30
Apakah member? (true/false): true
Biaya sewa setelah diskon (jika memenuhi syarat): Rp 8750.00
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> █
```

DEKSRIPSI:

Program ini menghitung biaya sewa berdasarkan durasi peminjaman (jam dan menit) serta status keanggotaan (member atau bukan). Durasi diubah ke jam desimal dan dikalikan tarif per jam: Rp 3500 untuk member, Rp 5000 untuk non-member. Fungsi hitungTotalJam sebelumnya salah karena selalu mengembalikan 2.5, dan perlu diperbaiki agar menghitung durasi dengan benar.

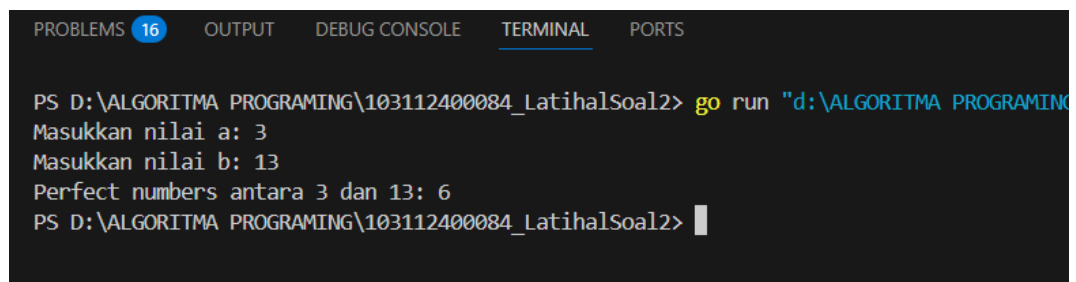
2

```
package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

func main() {
    var a, b int
    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)
    perfectNumber(a, b)
}

func perfectNumber(a, b int) {
    var jumlah int
    if a > b {
        return
    }
    for i := 1; i <= a; i++ {
        if a%i == 0 {
            if i != a {
                jumlah += i
            }
        }
    }
    if jumlah == a {
        fmt.Print(" ", jumlah)
    }
    perfectNumber(a+1, b)
}
```

OUTPUT:



The screenshot shows a Go IDE interface with a terminal window. The terminal displays the output of the program: "Masukkan nilai a: 3", "Masukkan nilai b: 13", and "Perfect numbers antara 3 dan 13: 6". The terminal prompt is "PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2>".

```
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2> go run "d:\ALGORITMA PROGRAMING
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2>
```

DEKSRIPSI:

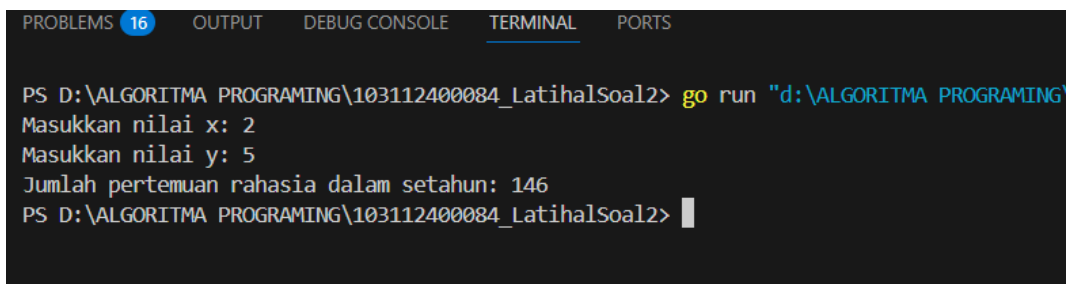
Program ini mencari dan menampilkan bilangan sempurna antara dua angka menggunakan rekursi. Bilangan sempurna adalah angka yang jumlah faktor selain dirinya sendiri sama dengan angka itu sendiri. Program menerima input a dan b, lalu memeriksa setiap angka dalam rentang tersebut secara rekursif.

3

```
package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

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

OUTPUT:



```
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan rahasia dalam setahun: 146
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> |
```

DEKSRIPSI:

Program ini menghitung jumlah hari pertemuan dalam setahun (365 hari) berdasarkan dua input bilangan bulat, x dan y. Fungsi rendezvous() menjumlahkan hari yang merupakan kelipatan x tetapi bukan kelipatan y, dengan melakukan iterasi dari 1 hingga 365 dan menghitung setiap hari yang memenuhi syarat tersebut.

tersebut terpenuhi, kemudian mencetak hasilnya sebagai jumlah pertemuan dalam setahun.

SOAL TIPE B

1

```
package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

func main() {
    var bil1, bil2, hasil int
    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&bil1)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&bil2)
    voucher(bil1, bil2, &hasil)
    fmt.Print("Banyaknya angka ganjil: ", hasil)
}

func voucher(a, b int, c *int) {
    for i := a; i <= b; i++ {
        if i%2 != 0 {
            *c += 1
        }
    }
}
```

OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2> go run "d:\ALGORITMA PROGRAMING\103112400084_LatihSoal2\main.go"
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2> |
```

DEKSRIPSI:

Program ini menghitung jumlah angka ganjil antara dua bilangan yang dimasukkan pengguna. Fungsi `voucher()` menggunakan perulangan untuk mengecek setiap angka dalam rentang `a` hingga `b`, dan menambah hasil setiap kali ditemukan angka ganjil. Hasil akhirnya ditampilkan sebagai total angka ganjil dalam rentang tersebut.

2

```
package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

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

func totalBayar(a int) {
    var jumlahMenu, banyakOrang, sisa, harga int
    for i := 1; i <= a; i++ {
        fmt.Print("Masukkan jumlah menu, jumlah orang, dan status sisa makanan
(0 untuk tidak, 1 untuk iya): ")
        fmt.Scan(&jumlahMenu, &banyakOrang, &sisa)
        if jumlahMenu > 50 {
            harga = 100000
        } else if jumlahMenu > 3 {
            harga = 10000 + 2500*jumlahMenu
        } else if jumlahMenu <= 3 {
            harga = 10000
        }

        if sisa == 1 {
            harga *= banyakOrang
        }
        fmt.Printf("Total biaya untuk rombongan %d: Rp %d \n", i, harga)
    }
}
```

OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2.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
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> |
```

DEKSRIPSI:

Program ini menghitung total biaya konsumsi untuk beberapa rombongan berdasarkan jumlah menu, jumlah orang, dan status sisa makanan. Untuk setiap rombongan, harga ditentukan berdasarkan jumlah menu, lalu dikalikan jumlah orang jika ada sisa makanan. Hasilnya adalah total biaya konsumsi yang ditampilkan untuk masing-masing rombongan.

3

```
package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

func main() {
    fmt.Println("Program menjumlahkan bilangan kelipatan 4")
    fmt.Println("Masukkan bilangan (akhiri dengan bilangan negatif):")
    total := startRecursion()
    fmt.Printf("Total jumlah bilangan kelipatan 4: %d\n", total)
}

func startRecursion() int {
    return recursiveSum(0)
}

func recursiveSum(accumulator int) int {
    var num int
    fmt.Scan(&num)
    if num < 0 {
        return accumulator
    }
    if num > 0 && num%4 == 0 {
        return recursiveSum(accumulator + num)
    }
    return recursiveSum(accumulator)
}
```

OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400084_Latihals
Program menjumlahkan bilangan kelipatan 4
Masukkan bilangan (akhiri dengan bilangan negatif):
1 2 3 4 5 -1
Total jumlah bilangan kelipatan 4: 4
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400084_Latihals
Program menjumlahkan bilangan kelipatan 4
Masukkan bilangan (akhiri dengan bilangan negatif):
1 2 3 4 5 -1
Total jumlah bilangan kelipatan 4: 4
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> █
```

DEKSRIPSI:

Program ini menghitung jumlah bilangan kelipatan 4 dari input pengguna dengan dua metode: iteratif dan rekursif. Fungsi `kelipatanIteratif()` menjumlahkan kelipatan 4 hingga pengguna memasukkan bilangan negatif, sedangkan `kelipatanRekursif()` melakukan hal yang sama secara rekursif. Hasil dari kedua pendekatan ditampilkan untuk dibandingkan.

SOAL TIPE C

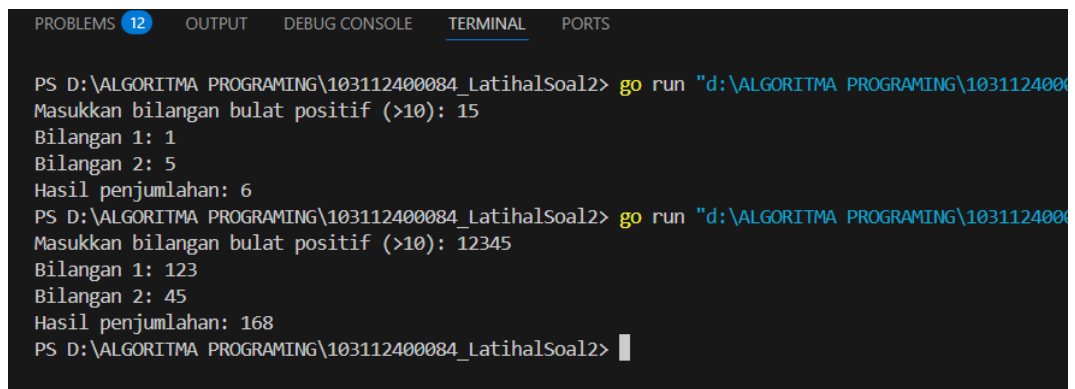
1

```
package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

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

func membagiDigit(a int) {
    var bil1, bil2, pembagi int
    pembagi = 1
    tempBil := a
    for a >= 10 {
        pembagi *= 10
        a = a / pembagi
    }
    if tempBil >= 100 {
        pembagi /= 10
    }
    bil1 = tempBil / pembagi
    bil2 = tempBil % pembagi
    fmt.Println("Bilangan 1:", bil1)
    fmt.Println("Bilangan 2:", bil2)
    fmt.Println("Hasil penjumlahan:", bil1+bil2)
}
```

OUTPUT:



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2> go run "d:\ALGORITMA PROGRAMING\103112400084_LatihSoal2\main.go"
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2> go run "d:\ALGORITMA PROGRAMING\103112400084_LatihSoal2\main.go"
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2>
```

DEKSRIPSI:

Program ini membagi sebuah bilangan bulat positif (lebih dari 10) menjadi dua bagian: digit pertama dan sisa digit di belakangnya, lalu menjumlahkan keduanya. Fungsi `membagiDigit()` menentukan pembagi untuk memisahkan digit pertama, lalu mencetak digit pertama (`bil1`), sisa digit (`bil2`), dan hasil penjumlahannya. Contoh: input 123 menghasilkan `bil1 = 1`, `bil2 = 23`, dan totalnya 24.

.

```

package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

func main() {
    var N, kode, jumA, jumB, jumC int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&N)
    for i := 1; i <= N; 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)
}

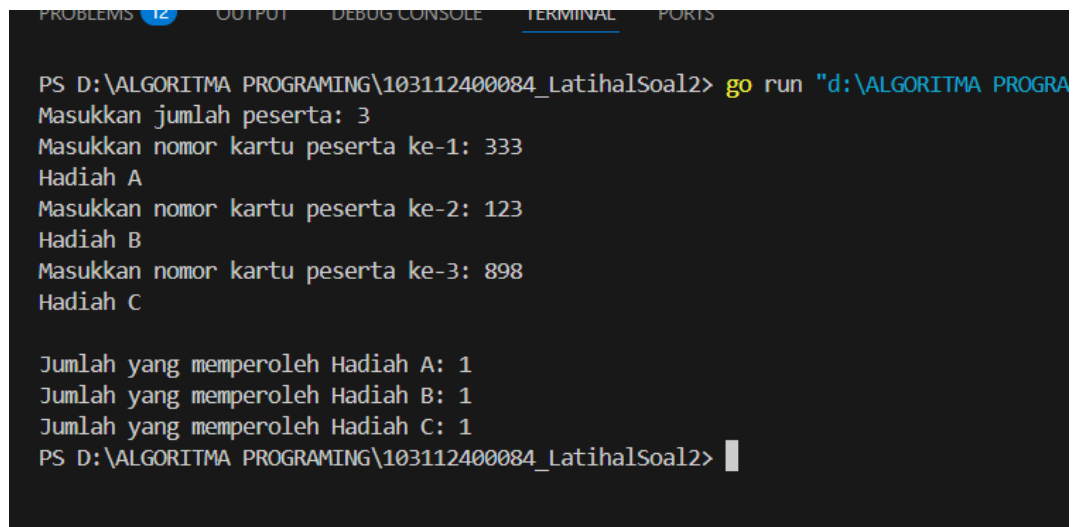
func bilanganSama(a int) bool {
    var b1, b2, b3 int
    b1 = a / 100
    b2 = (a / 10) % 10
    b3 = a % 10
    if b1 == b2 && b2 == b3 && b3 == b1 {
        return true
    }
    return false
}

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
}
```

OUTPUT:



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> go run "d:\ALGORITMA PROGRA
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:\ALGORITMA PROGRAMING\103112400084_LatihanSoal2> |
```

DEKSRIPSI:

Program ini menentukan jenis hadiah berdasarkan nomor kartu 3 digit yang dimasukkan oleh peserta. Jika semua digit sama, peserta mendapat Hadiah A; jika semua digit berbeda, mendapat Hadiah B; dan jika hanya sebagian digit yang sama, mendapat Hadiah C. Program menghitung dan menampilkan jumlah penerima untuk masing-masing jenis hadiah.

3

```
package main
//Nufail Alauddin Tsaqif
//103002400084
import "fmt"

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

func perkalianRekursif(n, m int) int {
    if n == 0 || m == 0 {
        return 0
    }
    if n == 1 {
        return m
    }
    if m == 1 {
        return n
    }

    // Rekursi tail-end
    if n > 0 {
        return m + perkalianRekursif(n-1, m)
    }
    // Untuk kasus bilangan negatif
    return -perkalianRekursif(-n, m)
}
```

OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2> go run
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Hasil dari 5 x 6 = 30
PS D:\ALGORITMA PROGRAMING\103112400084_LatihSoal2> |
```

DEKSRIPSI:

Program ini menghitung hasil perkalian dua bilangan ($n \times m$) tanpa menggunakan operator *, dengan dua pendekatan: iteratif dan rekursif. Metode iteratif menjumlahkan m sebanyak n kali dalam loop, sedangkan metode rekursif menambahkan m ke hasil secara berulang hingga n habis. Pengguna memasukkan dua bilangan, dan hasil dari kedua metode ditampilkan untuk menunjukkan konsep perkalian sebagai penjumlahan berulang.