

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



Oleh:

NAMA: MOHAMMAD REYHAN ARETHA FATIN

NIM: 103112400078

KELAS: IF-12-01

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## SOAL TIPE A

1

```
package main

//103112400078
//Mohammad Reyhan Aretha Fatin
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)

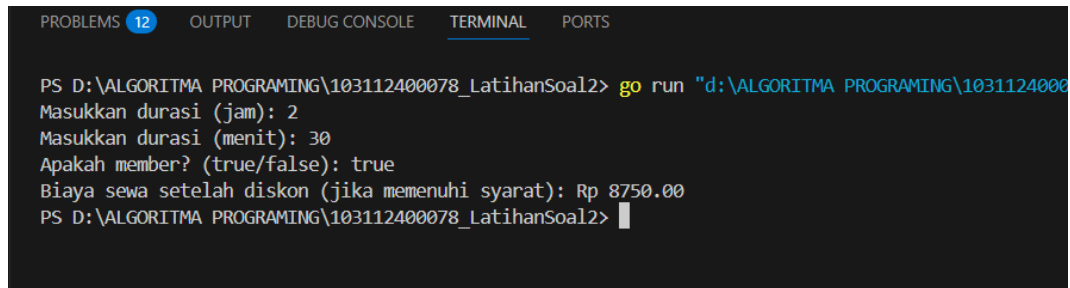
    fmt.Printf("Biaya sewa setelah diskon (jika memenuhi syarat): Rp %.2f\n",
biaya)
}

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 Go IDE terminal window. The terminal has tabs for PROBLEMS (12), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS. The terminal text shows the execution of a Go program. The user enters 'go run' followed by the file path. 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\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2\main.go"
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\103112400078_LatihanSoal2>
```

## DEKSRIPSI:

Program yang menghitung biaya sewa berdasarkan durasi peminjaman (jam dan menit) dan status keanggotaan pengguna (member atau bukan). Pengguna memasukkan durasi sewa dan status member, lalu program menghitung total durasi dalam jam desimal dan mengalikannya dengan tarif per jam (Rp 3500 untuk member, Rp 5000 untuk non-member). Namun, fungsi penghitungan total jam (`hitungTotalJam`) belum diimplementasikan dengan benar karena selalu mengembalikan nilai tetap 2.5, sehingga perlu diperbaiki agar menghitung durasi dengan tepat.


## 2

```
package main
//103112400078
//Mohammad Reyhan Aretha Fatin
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 following output:

```
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> |
```

#### DEKSRIPI:

Program ini adalah program rekursif yang mencari dan menampilkan bilangan sempurna (perfect number) di antara dua angka yang dimasukkan oleh pengguna (a dan b). Bilangan sempurna adalah bilangan yang jumlah faktor-faktornya (selain dirinya sendiri) sama dengan bilangan itu sendiri (contoh: 6, karena  $1 + 2 + 3 = 6$ ). Program meminta input a dan b, lalu memanggil fungsi perfectNumber secara rekursif dari a hingga b, mengecek setiap bilangan apakah merupakan bilangan sempurna dan mencetaknya jika iya.

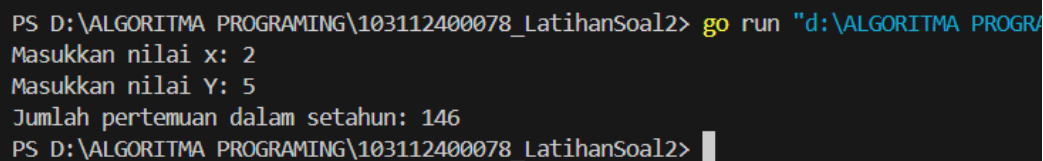
### 3

```
package main
//103112400078
//Mohammad Reyhan Aretha Fatin
import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan nilai Y: ")
    fmt.Scan(&y)
    fmt.Print("Jumlah pertemuan dalam setahun: ", rendezvous(x, y))
}

func rendezvous(x, y int) int {
    var jumlah int
    for i := 1; i <= 365; i++ {
        if i%x == 0 && i%y != 0 {
            jumlah += 1
        }
    }
    return jumlah
}
```

OUTPUT:



```
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRA
Masukkan nilai x: 2
Masukkan nilai Y: 5
Jumlah pertemuan dalam setahun: 146
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> █
```

DEKSRIPSI:

Program yang menghitung jumlah hari pertemuan dalam setahun (365 hari), berdasarkan dua nilai input x dan y. Program menerima dua bilangan bulat x dan y dari pengguna, lalu menggunakan fungsi rendezvous() untuk menghitung berapa hari yang merupakan kelipatan dari x tetapi bukan kelipatan dari y. Program ini melakukan iterasi dari 1 hingga 365 dan menambahkan jumlah setiap kali kondisi tersebut terpenuhi, kemudian mencetak hasilnya sebagai jumlah pertemuan dalam setahun.

## SOAL TIPE B

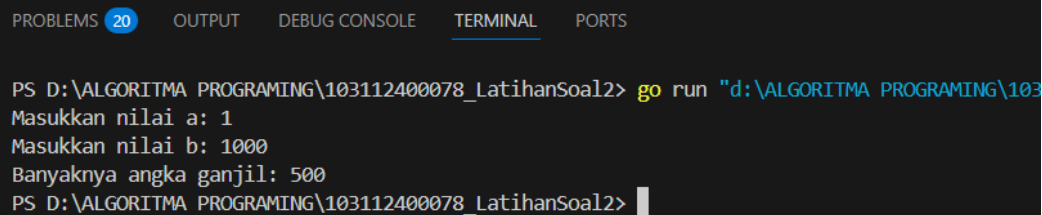
1

```
package main
//103112400078
//Mohammad Reyhan Aretha Fatin
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:



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS 20', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The terminal content shows the command 'go run "d:\ALGORITMA PROGRAMING\103112400078\_LatihanSoal2"' being executed. The output of the program is displayed as follows:

```
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2"
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2>
```

#### DEKSRIPI:

Program yang menghitung banyaknya angka ganjil di antara dua bilangan yang dimasukkan oleh pengguna (a dan b). Pengguna memberikan input dua bilangan (bil1 dan bil2), lalu program memanggil fungsi voucher() yang menerima pointer ke variabel hasil untuk menghitung berapa banyak angka ganjil dari a hingga b. Fungsi ini menggunakan perulangan dan menambahkan 1 ke nilai hasil setiap kali ditemukan bilangan ganjil ( $i \% 2 \neq 0$ ). Hasil akhirnya ditampilkan sebagai jumlah angka ganjil dalam rentang tersebut.



## 2

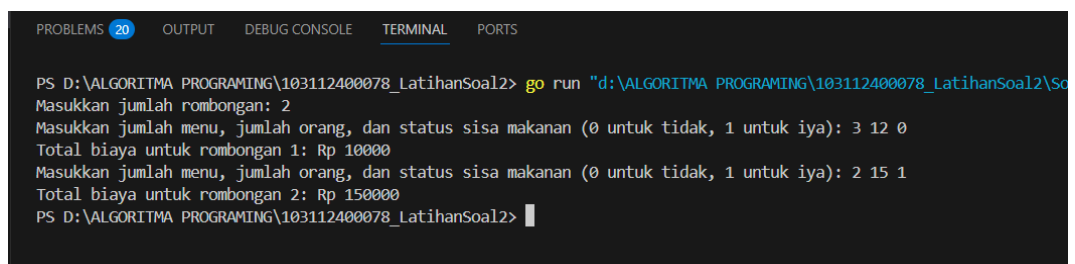
```
package main
//103112400078
//Mohammad Reyhan Aretha Fatin
import "fmt"

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\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2\Sc
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\103112400078_LatihanSoal2> █
```

#### DEKSRIPSI:

Program yang menghitung total biaya konsumsi untuk beberapa rombongan berdasarkan jumlah menu, jumlah orang, dan apakah ada sisa makanan. Pengguna memasukkan jumlah rombongan, lalu untuk tiap rombongan diminta input jumlah menu, jumlah orang, dan status sisa makanan (0 = tidak ada sisa, 1 = ada sisa). Program menentukan harga sesuai ketentuan: jika menu lebih dari 50 maka biaya tetap Rp100.000, jika lebih dari 3 maka dihitung Rp10.000 ditambah Rp2.500 per menu, dan jika 3 atau kurang hanya Rp10.000. Jika ada sisa makanan, harga dikalikan jumlah orang. Hasil akhir adalah total biaya tiap rombongan yang ditampilkan satu per satu.

### 3

```
package main
//103112400078
//Mohammad Reyhan Aretha Fatin
import "fmt"

func kelipatanIteratif() {
    var bilangan, hasil int = 0, 0
    fmt.Println("Masukkan bilangan (negatif untuk berhenti): ")
    for bilangan >= 0 {
        fmt.Scan(&bilangan)
        if bilangan%4 == 0 {
            hasil += bilangan
        }
    }
    fmt.Println("Iteratif--Jumlah bilangan kelipatan 4:", hasil)
}

func kelipatanRekursif(a int) int {
    var bil int
    fmt.Scan(&bil)
    if bil < 0 {
        return a
    }
    if bil >= 0 && bil%4 == 0 {
        a += bil
    }
    return kelipatanRekursif(a)
}

func main() {
    kelipatanIteratif()
    fmt.Println("Masukkan bilangan (negatif untuk berhenti): ")
    hasil := kelipatanRekursif(0)
    fmt.Print("Rekursif--Jumlah bilangan kelipatan 4: ", hasil)
}
```

## OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Iteratif--Jumlah bilangan kelipatan 4: 4
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Rekursif--Jumlah bilangan kelipatan 4: 4
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> █
```

## DEKSRIPSI:

Program yang menghitung jumlah bilangan kelipatan 4 dari input pengguna dengan dua cara berbeda: secara iteratif dan secara rekursif. Pertama, fungsi `kelipatanIteratif()` membaca bilangan satu per satu dari input hingga bilangan negatif dimasukkan, dan menjumlahkan hanya bilangan yang merupakan kelipatan 4. Kedua, fungsi `kelipatanRekursif()` melakukan hal yang sama tetapi dengan pendekatan rekursif: memanggil dirinya sendiri hingga input bernilai negatif, sambil menjumlahkan kelipatan 4. Hasil dari kedua metode ditampilkan secara terpisah, sehingga pengguna bisa melihat perbandingan hasil antara pendekatan iteratif dan rekursif.

## SOAL TIPE C

1

```
package main
//103112400078
//Mohammad Reyhan Aretha Fatin
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 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078_LatihanSoa
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078_LatihanSoa
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> |
```

#### DEKSRIPI:

Program yang membagi sebuah bilangan bulat positif (lebih dari 10) menjadi dua bagian: digit pertama (paling kiri) dan sisa digit setelahnya, lalu menjumlahkan keduanya. Pengguna diminta memasukkan sebuah bilangan, kemudian fungsi `membagiDigit()` akan mencari pembagi (pembagi) yang sesuai untuk memisahkan digit pertama dari sisanya. Nilai `bil1` berisi digit pertama (hasil pembagian), dan `bil2` berisi sisanya (hasil modulus). Akhirnya, program mencetak `bil1`, `bil2`, dan hasil penjumlahan keduanya. Contoh: jika input 123, maka `bil1 = 1`, `bil2 = 23`, dan hasilnya 24.

```

package main
//103112400078
//Mohammad Reyhan Aretha Fatin
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)
}

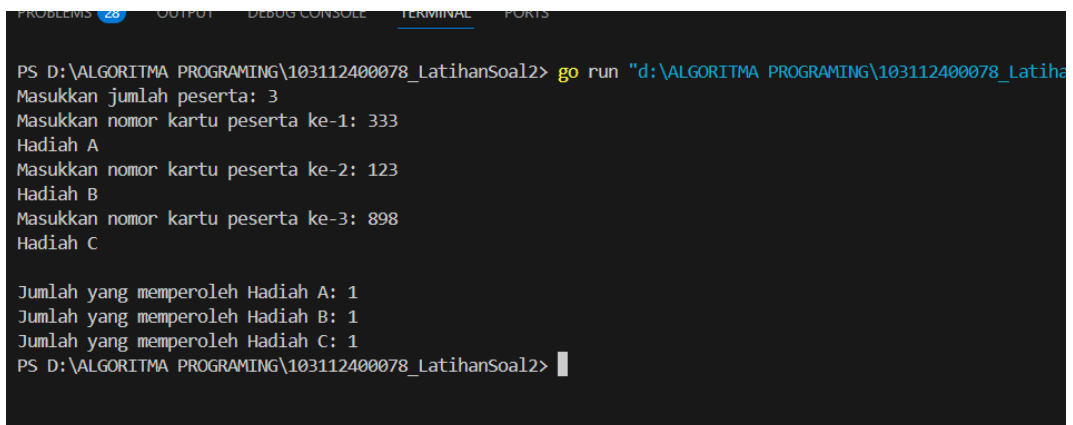
func bilanganSama(a int) bool {
    var b1, b2, b3 int
    b1 = a / 100
    b2 = (a / 10) % 10
    b3 = a % 10
    if b1 == b2 && b2 == b3 && b3 == b1 {
        return true
    }
    return false
}

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:



The screenshot shows a terminal window with the following text:

```
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078_Latih
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\103112400078_LatihanSoal2> |
```

DEKSRIPSI:

Program ini adalah aplikasi Go yang menentukan jenis hadiah yang diterima peserta berdasarkan nomor kartu 3 digit. Pengguna memasukkan jumlah peserta dan nomor kartu masing-masing, lalu program mengevaluasi digit-digit pada nomor tersebut. Jika semua digit sama, peserta mendapat Hadiah A; jika semua digit berbeda, mendapat Hadiah B; dan jika sebagian digit sama, mendapat Hadiah C. Program juga menghitung dan menampilkan jumlah peserta yang mendapatkan masing-masing jenis hadiah.



### 3

```
package main
//103112400078
//Mohammad Reyhan Aretha Fatin
import "fmt"

func perkalianIteratif(n, m int) int {
    var hasil int
    for i := n; i >= 1; i-- {
        hasil += m
    }
    return hasil
}

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

func main() {
    var n, m int
    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&n)
    fmt.Print("Masukkan bilangan m: ")
    fmt.Scan(&m)
    hasilIteratif := perkalianIteratif(n, m)
    hasilRekursif := perkalianRekursif(n, m, 0)
    fmt.Printf("Iteratif---Hasil dari %d x %d =%d\n", n, m, hasilIteratif)
    fmt.Printf("Rekursif---Hasil dari %d x %d =%d\n", n, m, hasilRekursif)
}
```

OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> go run "d:\ALGORITMA PROGRAMING\103112400078_
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Iteratif---Hasil dari 5 x 6 =30
Rekursif---Hasil dari 5 x 6 =30
PS D:\ALGORITMA PROGRAMING\103112400078_LatihanSoal2> |
```

## DEKSRIPSI:

Program ini adalah aplikasi Go yang melakukan perkalian dua bilangan ( $n \times m$ ) menggunakan dua pendekatan: iteratif dan rekursif, tanpa menggunakan operator perkalian langsung (\*). Pada pendekatan iteratif, program menjumlahkan  $m$  sebanyak  $n$  kali dalam loop. Pada pendekatan rekursif, program terus menambahkan  $m$  ke hasil hingga  $n$  mencapai 0, menggunakan pemanggilan fungsi berulang. Pengguna diminta memasukkan dua bilangan ( $n$  dan  $m$ ), lalu hasil dari kedua metode ditampilkan sebagai perbandingan. Program ini membantu memahami konsep perkalian sebagai penjumlahan berulang dan perbandingan antara rekursi dan iterasi.