

LATIHAN SOAL 2

ALGORITMA DAN PEMROGRAMAN 2



Oleh :

NAMA : Felix Pedrosa Valentino

NIM : 103112400056

KELAS : IF – 12 – 01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. Soal Tipe A

1. Sewa Sepeda

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

// Subprogram untuk menghitung biaya sewa
func hitungBiayaSewa(jam int, menit int, member bool, noVoucher string)
float64 {

    // Konstanta tarif
    var tarif float64

    if member {
        tarif = 3500
    } else {
        tarif = 5000
    }

    // Hitung total durasi dalam jam
    totalMeni := jam*60 + menit
    var totalJam int

    if totalMeni%60 >= 10 {
        totalJam = totalMeni/60 + 1
    } else {
        totalJam = totalMeni / 60
    }
}
```

```
// Hitung biaya sebelum diskon
biaya := float64(totalJam) * tarif

// Cek memenuhi syarat diskon
if len(noVoucher) >= 5 && len(noVoucher) <= 6 {
    diskon := 0.1 * biaya
    biaya -= diskon
}

return biaya
}

func main() {
    // Input
    var jam, menit int
    var member_103112400056 bool
    var noVoucher 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_103112400056)
    fmt.Print("Masukkan nomor voucher (jika ada): ")
    fmt.Scan(&noVoucher)
```

```
// Hitung biaya sewa

biaya := hitungBiayaSewa(jam, menit, member_103112400056,
noVoucher)

// Output

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

}
```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeA\SewaSepeda\SewaSepeda.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 9450.00
```

Penjelasan Program :

Program diatas ditulis dalam bahasa Go dan berfungsi untuk menghitung biaya sewa berdasarkan durasi penggunaan, status keanggotaan, dan penggunaan voucher diskon. Pengguna akan diminta untuk memasukkan durasi sewa dalam jam dan menit, serta menentukan apakah mereka merupakan anggota atau bukan.

Bagi anggota, tarif sewa ditetapkan sebesar Rp 3. 500 per jam, sedangkan bagi non-anggota tarifnya adalah Rp 5. 000 per jam. Selain itu, program ini juga memeriksa syarat penggunaan voucher diskon, yang memberikan potongan 10% jika nomor voucher yang dimasukkan memenuhi ketentuan panjang antara 5 hingga 6 karakter. Setelah semua input diterima, program akan menghitung dan menampilkan total biaya sewa, dengan penyesuaian diskon jika memenuhi syarat.

2. Perfect Number

Source Code :

```
// Felix Pedrosa V
```

```
package main

import "fmt"

// Fungsi untuk memeriksa apakah suatu bilangan adalah perfect number
atau tidak
func isPerfectNumber(n int) bool {
    sum := 0
    for i := 1; i <= n/2; i++ {
        if n%i == 0 {
            sum += i
        }
    }
    return sum == n
}

// Fungsi untuk mencetak perfect numbers dalam rentang a hingga b
func printPerfectNumbers(a, b int) {
    fmt.Printf("Perfect numbers antara %d dan %d: ", a, b)
    for i := a; i <= b; i++ {
        if isPerfectNumber(i) {
            fmt.Printf("%d ", i)
        }
    }
    fmt.Println() // Untuk baris baru setelah output
}

func main() {
```

```

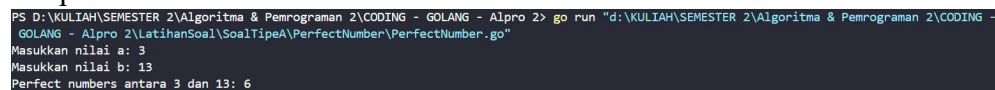
var a, b_103112400056 int

// Input
fmt.Print("Masukkan nilai a: ")
fmt.Scan(&a)
fmt.Print("Masukkan nilai b: ")
fmt.Scan(&b_103112400056)

// Validasi input agar a <= b
if a <= b_103112400056 {
    printPerfectNumbers(a, b_103112400056)
} else {
    fmt.Println("Nilai a harus kurang dari atau sama dengan b.")
}
}

```

Output :



```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeA\PerfectNumber\PerfectNumber.go"
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa pemrograman Go dan berfungsi untuk menghitung bilangan sempurna dalam rentang yang ditentukan oleh pengguna. Terdapat dua fungsi utama dalam program ini: fungsi `isPerfectNumber` , yang bertugas untuk memeriksa apakah suatu bilangan merupakan bilangan sempurna, yakni bilangan yang sama dengan jumlah semua pembaginya, dan fungsi `printPerfectNumbers` , yang mencetak semua bilangan sempurna di antara dua angka yang diberikan.

Pengguna akan diminta untuk memasukkan dua bilangan bulat, yaitu a dan b. Sebelum melanjutkan pencarian dan pencetakan bilangan sempurna dalam rentang tersebut, program akan memvalidasi bahwa a

kurang dari atau sama dengan b. Jika tidak, program akan memberikan pesan “Nilai a harus kurang dari atau sama dengan b.” agar pengguna dapat memperbaiki inputnya.

3. Rendezvous

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

func main() {
    var x_103112400056, y int

    // Input values for x and y
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x_103112400056)
    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y)

    // Counting meetings days
    jumlahPertemuan := 0
    // Iterate from 1 to 365 to find meeting days
    for hari := 1; hari <= 365; hari++ {
        if hari%x_103112400056 == 0 && hari%y != 0 {
            jumlahPertemuan++
        }
    }

    // Display result
    fmt.Printf("Jumlah pertemuan dalam setahun: %d\n",
jumlahPertemuan)
}
```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeA\Rendezvous\Rendezvous.go"
Masukkan nilai x: 2
Masukkan nilai y: 5
Jumlah pertemuan dalam setahun: 146
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan berfungsi untuk menghitung jumlah pertemuan rahasia antara dua agen mata-mata dalam rentang waktu satu tahun (365 hari). Pengguna akan diminta untuk memasukkan dua bilangan bulat positif, yaitu x dan y . Program kemudian akan menghitung jumlah hari di mana pertemuan berlangsung, yaitu pada hari-hari yang merupakan kelipatan dari x tetapi bukan kelipatan dari y . Proses perhitungan dilakukan dengan melakukan iterasi dari hari pertama hingga hari ke-365, dan hasilnya akan ditampilkan sebagai output.

II. Soal Tipe B

1. Voucher

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

func main() {
    var a, b_103112400056 int

    // Meminta input dari pengguna
    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b_103112400056)

    // Validasi input
    if a > b_103112400056 {
        fmt.Println("Kesalahan: Nilai a harus lebih kecil atau sama dengan b.")
        return
    }

    // Menghitung jumlah angka ganjil dalam rentang [a, b]
    count := 0
    for i := a; i <= b_103112400056; i++ {
```

```

        if i%2 != 0 {
            count++
        }
    }

    // Menampilkan hasil

    fmt.Printf("Banyaknya angka ganjil: %d", count)
}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeB\Voucher\Voucher.go"
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk menghitung serta menampilkan jumlah angka ganjil dalam rentang antara dua bilangan bulat positif, yaitu a dan b. Pengguna akan diminta untuk memasukkan nilai a dan b, dengan syarat a harus lebih kecil atau sama dengan b. Setelah menerima input, program akan melakukan validasi untuk memastikan bahwa nilai yang dimasukkan benar. Jika validasi berhasil, program akan menghitung jumlah angka ganjil dalam rentang tersebut dan menampilkan hasilnya. Sebagai contoh, jika pengguna memasukkan nilai a sebagai 1 dan b sebagai 1000, program akan menunjukkan bahwa terdapat 500 angka ganjil dalam rentang tersebut.

2. Restoran

Source Code :

```

// Felix Pedrosa V

package main

```

```
import "fmt"

func hitungBiaya(jumlahMenu int) int {
    if jumlahMenu <= 3 {
        return 10000
    } else if jumlahMenu > 50 {
        return 100000
    } else {
        return 10000 + (jumlahMenu-3)*2500
    }
}

func main() {
    var rombongan_103112400056 int
    fmt.Print("Masukkan jumlah rombongan: ")
    fmt.Scan(&rombongan_103112400056)

    totalBiaya := 0

    for i := 1; i <= rombongan_103112400056; i++ {
        var jumlahMenu, jumlahOrang, sisaMakanan int

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

        biaya := hitungBiaya(jumlahMenu)
```

```

        if sisaMakanan == 1 {
            biaya *= jumlahOrang
        }

        totalBiaya += biaya

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

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeB\Restoran\Restoran.go"
Masukkan jumlah rombongan: 2
Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya) 1: 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: 2 15 1
Total biaya untuk rombongan 2: Rp 150000

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk menghitung total biaya berdasarkan jumlah menu yang dipesan oleh setiap rombongan. Pengguna akan diminta untuk memasukkan jumlah rombongan, diiringi dengan informasi yang relevan untuk setiap rombongan, yaitu jumlah menu yang dipesan, jumlah anggota dalam rombongan, dan status sisa makanan.

Fungsi hitungBiaya akan melakukan perhitungan biaya sesuai dengan skema harga yang telah ditetapkan, jika jumlah menu yang dipesan adalah 3 atau kurang, biayanya tetap Rp 10.000; jika jumlah menu lebih dari 50, biayanya menjadi Rp 100.000; dan untuk jumlah menu antara 4 hingga 50, biaya akan dihitung dengan menambahkan biaya dasar untuk setiap menu tambahan. Apabila terdapat sisa makanan, total biaya tersebut akan dikalikan dengan jumlah anggota dalam rombongan. Setelah semua informasi dimasukkan, program akan menampilkan total biaya untuk setiap rombongan.

3. Jumlah Bilangan Positif Kelipatan 4

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

// Fungsi rekursif untuk menjumlahkan bilangan positif kelipatan 4
func jumlahKelipatanEmpat(angka []int, indeks int) int {
    // Jika sudah mencapai akhir array atau bilangan negatif (sentinel),
    kembalikan 0
    if indeks >= len(angka) || angka[indeks] < 0 {
        return 0
    }

    // Cek jika bilangan adalah positif kelipatan 4, tambahkan ke hasil
    if angka[indeks]%4 == 0 && angka[indeks] > 0 {
        return angka[indeks] + jumlahKelipatanEmpat(angka, indeks+1)
    } else {
        // Jika bukan kelipatan 4, lanjutkan ke bilangan berikutnya
        return jumlahKelipatanEmpat(angka, indeks+1)
    }
}

func main() {
    var input_103112400056 int
    angka := []int{}
```

```

fmt.Println("Masukkan bilangan (negatif untuk berhenti):")

// Ambil input dari pengguna hingga bilangan negatif dimasukkan
for {
    fmt.Scan(&input_103112400056)
    if input_103112400056 < 0 {
        break
    }
    angka = append(angka, input_103112400056)
}

// Hitung jumlah bilangan kelipatan 4 menggunakan fungsi rekursif
hasil := jumlahKelipatanEmpat(angka, 0)

// Tampilkan hasil
fmt.Printf("Jumlah bilangan kelipatan 4: %d\n", hasil)
}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSeal\SealTipe8\JmlhBilPositifKelipatan4\JmlhBilPositifKelipatan4.go"
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Jumlah bilangan kelipatan 4: 4

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk menghitung jumlah bilangan bulat positif yang merupakan kelipatan 4 berdasarkan input dari pengguna. Dengan pendekatan rekursif, program ini mampu menjumlahkan semua kelipatan 4 yang dimasukkan. Pengguna diminta untuk memberikan serangkaian bilangan bulat, dan proses akan berakhir ketika pengguna memasukkan bilangan negatif sebagai penanda. Fungsi `jumlahKelipatanEmpat` akan secara rekursif memeriksa setiap bilangan dalam array yang diberikan, menjumlahkan hanya bilangan

positif yang merupakan kelipatan 4, sambil mengabaikan bilangan negatif. Setelah itu, hasil total dari jumlah bilangan kelipatan 4 tersebut akan ditampilkan kepada pengguna.

III. Soal Tipe C

1. Digit

Source Code :

```
// Felix Pedrosa V

package main

import (
    "fmt"
    "strconv"
)

func main() {
    var input_103112400056 string
    fmt.Print("Masukkan bilangan bulat positif (>10): ")
    fmt.Scan(&input_103112400056)

    num, err := strconv.Atoi(input_103112400056)
    if err != nil || num <= 10 {
        fmt.Println("Input tidak valid, harus lebih besar dari 10.")
        return
    }

    // Mendapatkan panjang digit
    length := len(input_103112400056)

    var bil1, bil2 string
```



```

if length%2 == 0 {
    mid := length / 2
    bil1 = input_103112400056[:mid]
    bil2 = input_103112400056[mid:]
} else {
    mid := length / 2
    bil1 = input_103112400056[:mid+1]
    bil2 = input_103112400056[mid+1:]
}

bil1Int, _ := strconv.Atoi(bil1)
bil2Int, _ := strconv.Atoi(bil2)
sum := bil1Int + bil2Int

// Menampilkan hasil
fmt.Println("Bilangan 1:", bil1)
fmt.Println("Bilangan 2:", bil2)
fmt.Println("Hasil penjumlahan:", sum)
}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeC\Digit\Digit.go"
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeC\Digit\Digit.go"
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk memecah sebuah bilangan bulat positif yang dimasukkan oleh pengguna menjadi dua bagian berdasarkan panjang digitnya. Pengguna diminta untuk memasukkan bilangan bulat positif yang lebih besar dari 10. Jika bilangan tersebut memenuhi syarat, program akan memeriksa panjang digitnya. Apabila panjang digitnya genap, bilangan akan dibagi menjadi dua bagian yang sama panjang. Sebaliknya, jika panjang digitnya ganjil, bagian pertama akan memiliki satu digit lebih banyak dibandingkan bagian kedua. Setelah itu, program akan menampilkan kedua bagian yang telah dipotong beserta hasil penjumlahan kedua bagian tersebut.

2. Bukber IF

Source Code :

```
package main

import (
    "fmt"
    "strconv"
)

// Fungsi untuk memeriksa apakah semua digit sama
func semuaDigitSama(n int) bool {
    nStr := strconv.Itoa(n)
    digit := nStr[0]
    for i := 1; i < len(nStr); i++ {
        if nStr[i] != digit {
            return false
        }
    }
    return true
}
```

```
// Fungsi untuk memeriksa apakah semua digit berbeda
func semuaDigitBerbeda(n int) bool {
    digitMap := make(map[rune]bool)
    nStr := strconv.Itoa(n)
    for _, digit := range nStr {
        if digitMap[digit] {
            return false
        }
        digitMap[digit] = true
    }
    return true
}

func main() {
    var n_103112400056 int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&n_103112400056)

    hadiah := make([]string, n_103112400056)
    hadiahA, hadiahB, hadiahC := 0, 0, 0

    for i := 0; i < n_103112400056; i++ {
        var kartu int
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i+1)
        fmt.Scan(&kartu)

        if semuaDigitSama(kartu) {
```

```

        hadiahhs[i] = "Hadiah A"

        hadiahA++
    } else if semuaDigitBerbeda(kartu) {
        hadiahhs[i] = "Hadiah B"
        hadiahB++
    } else {
        hadiahhs[i] = "Hadiah C"
        hadiahC++
    }
}

for _, hadiah := range hadiahhs {
    fmt.Println(hadiah)
}

fmt.Printf("Jumlah 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 D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeC\BukberIF\BukberIF.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

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk menentukan jenis hadiah yang akan diterima oleh peserta dalam lomba

kelereng yang diselenggarakan oleh mahasiswa IF Lab Universitas Telkom. Setiap peserta diminta untuk memasukkan nomor kartu unik yang terdiri dari angka positif. Program ini memiliki dua fungsi utama: yang pertama, "semuaDigitSama", berfungsi untuk memeriksa apakah semua digit dalam nomor kartu peserta adalah sama; yang kedua, "semuaDigitBerbeda", untuk memastikan apakah semua digitnya berbeda. Berdasarkan hasil pemeriksaan ini, peserta akan menerima salah satu dari tiga jenis hadiah: Hadiah A untuk mereka yang nomor kartunya memiliki semua digit yang sama, Hadiah B untuk nomor kartu dengan semua digit yang berbeda, dan Hadiah C bagi nomor kartu yang tidak memenuhi kedua kriteria tersebut. Setelah semua peserta memasukkan nomor kartu mereka, program akan menampilkan jenis hadiah yang diterima oleh masing-masing peserta serta total jumlah peserta yang memperoleh setiap jenis hadiah.

3. Perkalian dengan Cara Penjumlahan

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

// Fungsi rekursif dengan tail-end recursion untuk menghitung perkalian
func perkalianRekursif(n, m, hasil int) int {
    if m == 0 {
        return hasil
    }
    return perkalianRekursif(n, m-1, hasil+n)
}

func main() {
    var n, m_103112400056 int
```

```

    fmt.Print("Masukkan bilangan n: ")

    fmt.Scan(&n)

    fmt.Print("Masukkan bilangan m: ")

    fmt.Scan(&m_103112400056)


    hasil := perkalianRekursif(n, m_103112400056, 0)

    fmt.Printf("Hasil dari %d x %d = %d\n", n, m_103112400056, hasil)

}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\LatihanSoal\SoalTipeC\perkalianAndcaraPenjumlahan\perkalianAndcaraPenjumlahan.go"
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Hasil dari 5 x 6 = 30

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk menghitung hasil perkalian dua bilangan bulat, yaitu n dan m, dengan menggunakan algoritma rekursif berbasis penjumlahan. Proses perhitungan dilakukan melalui fungsi yang disebut perkalianRekursif, yang menerapkan teknik tail-end recursion, di mana n akan ditambahkan sebanyak m kali.

Program dimulai dengan meminta pengguna untuk memasukkan nilai n dan m. Selanjutnya, fungsi rekursif akan dipanggil dengan hasil awal 0. Setelah seluruh proses selesai, program akan menampilkan hasil perkalian dengan format yang jelas. Sebagai contoh, jika pengguna memasukkan 5 untuk n dan 6 untuk m, maka output yang dihasilkan adalah Hasil dari 5 x 6 = 30 .