

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



Oleh:

NAMA: LUTFI SHIDQI MARDIAN

NIM: 103112400077

KELAS: IF 12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. SOAL TIPE A

KODE PROGRAM NO 1 (SEPEDA):

```
package main

import "fmt"

func tarif(jam, menit int, member bool, voucher int) float64{
    var tarifJam float64

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

    totalJam := float64(jam)

    if jam == 0 && menit > 0 {
        totalJam = 1
    } else if menit >= 10 {
        totalJam += float64(menit) / 60
    }

    totalTarif := totalJam * tarifJam

    if (voucher >= 10000 && voucher <= 99999) || (voucher >=
100000 && voucher <= 999999) {
        totalTarif *= 0.9
    }

    return totalTarif
}

func main() {
```

```

var jam_103112400077, menit, voucher int

var member bool

fmt.Print("Masukkan durasi (jam): ")

fmt.Scan(&jam_103112400077)

fmt.Print("Masukkan durasi (menit): ")

fmt.Scan(&menit)

fmt.Print("Apakah member? (true/false): ")

fmt.Scan(&member)

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

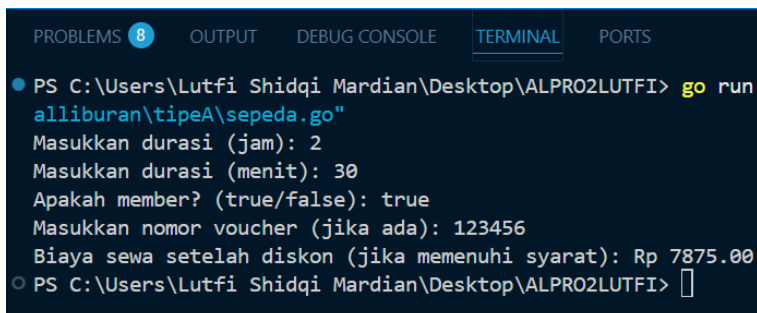
fmt.Scan(&voucher)

biaya := tarif(jam_103112400077, menit, member, voucher)

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

```

SS OUTPUT:



The screenshot shows a terminal window with the following output:

```

PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
alliburan\tipeA\sepeda.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 7875.00
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>

```

DESKRIPSI:

Program Go di atas digunakan untuk menghitung biaya sewa berdasarkan durasi waktu, status keanggotaan (member atau bukan), dan keberadaan voucher diskon. Fungsi tarif menerima input berupa jam, menit, status member, dan nomor voucher, lalu menentukan tarif per jam sebesar Rp3.500 untuk member dan Rp5.000 untuk non-member. Jika waktu sewa hanya beberapa menit namun lebih dari nol, maka dianggap 1 jam; jika lebih dari 10 menit, akan dihitung secara proporsional. Jika nomor voucher berada dalam rentang lima atau enam digit, maka pengguna mendapatkan diskon 10%. Program utama (main) meminta input dari pengguna melalui terminal dan mencetak total biaya sewa setelah mempertimbangkan diskon.

KODE PROGRAM NO 2(PERFECT NUMBER):

```
package main

import "fmt"

func perfek(n int) bool {
    sum := 1

    for i := 2; i*i <= n; i++ {
        if n%i == 0 {
            sum += i

            if i != n/i {
                sum += n / i
            }
        }
    }

    return sum == n && n != 1
}

func main() {
    var a_103112400077, b int

    fmt.Print("Masukkan nilai a: ")

    fmt.Scan(&a_103112400077)

    fmt.Print("Masukkan nilai b: ")

    fmt.Scan(&b)

    fmt.Printf("Perfect numbers antara %d dan %d:",
a_103112400077, b)

    first := true

    for i := a_103112400077; i <= b; i++ {
```

```

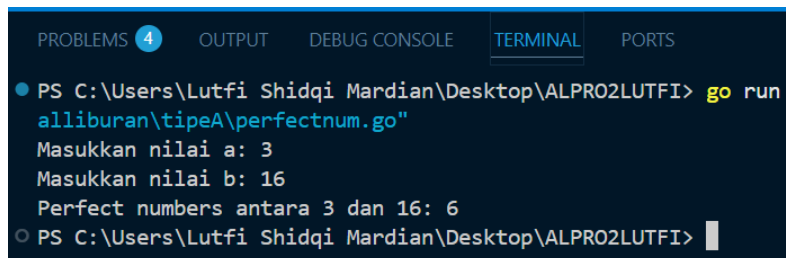
        if perfek(i) {
            if !first {
                fmt.Print(",")
            }
            fmt.Printf(" %d", i)
            first = false
        }
    }

    if first {
        fmt.Print(" Tidak ada")
    }

    fmt.Println()
}

```

SS OUTPUT:



```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
  alliburan\tipeA\perfectnum.go
  Masukkan nilai a: 3
  Masukkan nilai b: 16
  Perfect numbers antara 3 dan 16: 6
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>

```

DESKRIPSI:

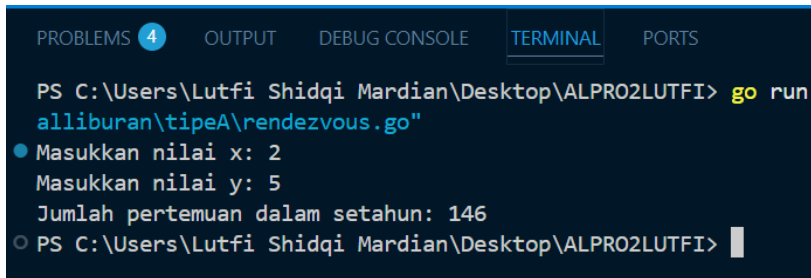
Program Go di atas digunakan untuk mencari dan menampilkan bilangan sempurna (perfect number) dalam rentang dari a hingga b yang dimasukkan oleh pengguna. Bilangan sempurna adalah bilangan yang jumlah faktor-faktor pembaginya (selain dirinya sendiri) sama dengan bilangan itu sendiri, contohnya 6 dan 28. Fungsi perfek mengecek apakah suatu bilangan n adalah bilangan sempurna dengan menjumlahkan semua faktor pembaginya yang sesuai. Di dalam fungsi main, program meminta input nilai a dan b, lalu mencetak semua bilangan sempurna di antara kedua nilai tersebut. Jika tidak ada bilangan sempurna dalam rentang tersebut, maka akan ditampilkan pesan "Tidak ada".

KODE PROGRAM NO 3(RENDEZVOUS):

```
package main
import "fmt"
func pertemuan(x, y int) int {
    jumlah := 0
    for hari := 1; hari <= 365; hari++ {
        if hari%x == 0 && hari%y != 0 {
            jumlah++
        }
    }
    return jumlah
}
func main() {
    var x_10311240077, y int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x_10311240077)
    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y)

    hasil := pertemuan(x_10311240077, y)
    fmt.Printf("Jumlah pertemuan dalam setahun: %d\n", hasil)
}
```

SS OUTPUT:

A screenshot of a terminal window with a dark blue background. At the top, there are tabs for 'PROBLEMS' (with a blue circle containing the number 4), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and highlighted in light blue), and 'PORTS'. The terminal shows the command 'go run alliburan\tipeA\rendezvous.go' being executed. The output consists of three lines: 'Masukkan nilai x: 2', 'Masukkan nilai y: 5', and 'Jumlah pertemuan dalam setahun: 146'. The prompt 'PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>' is visible at the bottom.

DESKRIPSI:

Program Go di atas digunakan untuk menghitung jumlah hari dalam satu tahun (365 hari) di mana suatu kegiatan yang terjadi setiap x hari berlangsung, tetapi tidak bertepatan dengan kegiatan lain yang terjadi setiap y hari. Fungsi pertemuan memeriksa setiap hari dari 1 hingga 365, menambahkan jumlah hari yang merupakan kelipatan dari x namun bukan kelipatan dari y. Di bagian main, program meminta input dari pengguna untuk nilai x dan y, lalu menampilkan jumlah hari pertemuan yang memenuhi syarat tersebut dalam satu tahun.

II. SOAL TIPE B

KODE PROGRAM NO 1(GANJIL DALAM RENTANG):

```
package main

import "fmt"

func ganjilRentang(a, b int) int {

    bilB := (b + 1) / 2

    bilA := (a - 1) / 2

    return bilB - bilA

}

func main() {

    var a_103112400077, b int

    fmt.Print("Masukkan nilai a: ")

    fmt.Scan(&a_103112400077)


    fmt.Print("Masukkan nilai b: ")

    fmt.Scan(&b)

    fmt.Print("Banyaknya angka ganjil: ", ganjilRentang(a_103112400077,
b))

}
```

SS OUTPUT:



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
alliburan\tipe8\rentang.go
Masukkan nilai a: 1
Masukkan nilai b: 1000
Banyaknya angka ganjil: 500
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>
```

DESKRIPSI:

Program Go di atas digunakan untuk menghitung jumlah bilangan ganjil dalam rentang dari a hingga b, termasuk a dan b jika keduanya ganjil. Fungsi ganjilRentang memanfaatkan rumus matematika sederhana dengan cara menghitung jumlah bilangan ganjil dari 1 sampai b, lalu dikurangi dengan jumlah bilangan ganjil dari 1 sampai a-1, sehingga didapatkan jumlah bilangan ganjil dalam rentang [a, b]. Di bagian main, program meminta input nilai a dan b dari pengguna, lalu mencetak hasil perhitungan jumlah bilangan ganjil dalam rentang tersebut.

KODE PROGRAM NO 2(RESTORAN):

```
package main

import "fmt"

func hitungBiaya(menu, orang, sisa int) int {

    biaya := 0

    if menu <= 3 {

        biaya = 10000

    } else if menu <= 50 {

        biaya = 10000 + (menu-3)*2500

    } else {

        biaya = 100000

    }

    if sisa == 1 {

        biaya *= orang

    }

    return biaya

}

func main() {

    var M_103112400077 int

    fmt.Print("Masukkan jumlah rombongan: ")

    fmt.Scan(&M_103112400077)

    for i := 1; i <= M_103112400077; i++ {

        var menu, orang, sisa int

        fmt.Printf("Masukkan jumlah menu, jumlah orang, dan status sisa makanan untuk rombongan %d: ", i)

        fmt.Scan(&menu, &orang, &sisa)
```



```

        totalBiaya := hitungBiaya(menu, orang, sisa)

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

```

SS OUTPUT:

```

PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run "c:\Users\Lutfi Shidqi Mard
alliburan\tipeB\restoran.go"
● Masukkan jumlah rombongan: 2
Masukkan jumlah menu, jumlah orang, dan status sisa makanan untuk rombongan 1: 3 12 0
Total biaya untuk rombongan 1: Rp 10000
Masukkan jumlah menu, jumlah orang, dan status sisa makanan untuk rombongan 2: 2 15 1
Total biaya untuk rombongan 2: Rp 150000
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>

```

DESKRIPSI:

Program Go di atas digunakan untuk menghitung **total biaya konsumsi** beberapa rombongan berdasarkan jumlah menu yang dipesan, jumlah orang dalam rombongan, dan status sisa makanan. Fungsi `hitungBiaya` menentukan biaya berdasarkan jumlah menu: untuk 3 menu atau kurang, biayanya Rp10.000; untuk lebih dari 3 hingga 50 menu, ditambah Rp2.500 per menu setelah menu ketiga; dan jika lebih dari 50 menu, biayanya flat Rp100.000. Jika ada sisa makanan (ditandai dengan `sisa == 1`), maka biaya akan dikalikan dengan jumlah orang sebagai bentuk penalti. Di fungsi `main`, program meminta jumlah rombongan dari pengguna, lalu untuk setiap rombongan, meminta data menu, jumlah orang, dan status sisa makanan, kemudian mencetak total biaya masing-masing rombongan.

KODE PROGRAM NO 3(KELIPATAN 4):

```
package ma

import "fmt"

func jumlahKelipatan4(total int) int {

    var angka_103112400077 int

    fmt.Scan(&angka_103112400077)

    if angka_103112400077 < 0 {

        return total

    }

    if angka_103112400077%4 == 0 {

        return jumlahKelipatan4(total + angka_103112400077)

    }

    return jumlahKelipatan4(total)

}

func main() {

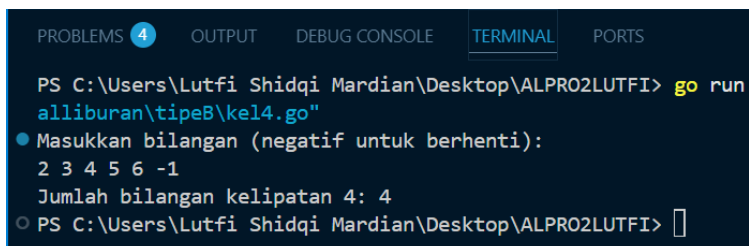
    fmt.Println("Masukkan bilangan (negatif untuk berhenti):")

    hasil := jumlahKelipatan4(0)

    fmt.Println("Jumlah bilangan kelipatan 4:", hasil)

}
```

SS OUTPUT:



The screenshot shows a terminal window with the following content:

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
alliburan\tipe8\kel4.go
● Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Jumlah bilangan kelipatan 4: 4
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> 
```

DESKRIPSI:

Program Go di atas digunakan untuk menghitung jumlah total dari bilangan kelipatan 4 yang dimasukkan oleh pengguna, hingga pengguna memasukkan bilangan negatif sebagai penanda berhenti. Fungsi jumlahKelipatan4 bekerja secara rekursif, menerima parameter total untuk menyimpan akumulasi jumlah. Setiap bilangan yang dimasukkan akan diperiksa: jika bilangan negatif, proses berhenti dan total dikembalikan; jika bilangan adalah kelipatan 4, maka nilainya ditambahkan ke total dan fungsi dipanggil lagi; jika bukan kelipatan 4, fungsi tetap dipanggil kembali tanpa menambah total. Di fungsi main, program meminta input berulang-ulang dari pengguna dan mencetak total bilangan kelipatan 4 yang berhasil dijumlahkan.

III. SOAL TIPE C

KODE PROGRAM NO 1(DIGIT):

```
package main

import "fmt"

func main() {

    var n_103112400077 int

    fmt.Print("Masukkan bilangan bulat positif (>10): ")

    fmt.Scan(&n_103112400077)

    temp := n_103112400077

    pjg := 0

    for temp > 0 {

        temp /= 10

        pjg++

    }

    var pisah int = 1

    for i := 0; i < pjg/2; i++ {

        pisah *= 10

    }

    if pjg%2 != 0 {

        pisah *= 10

    }

    kiri := n_103112400077 / pisah

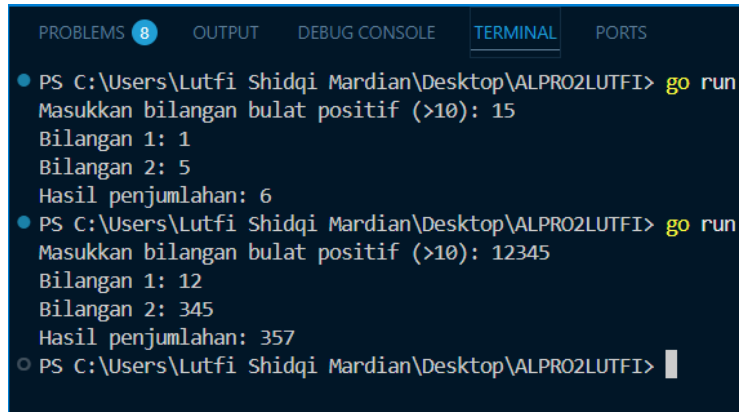
    kanan := n_103112400077 % pisah

    fmt.Println("Bilangan 1:", kiri)

    fmt.Println("Bilangan 2:", kanan)
```

```
    fmt.Println("Hasil penjumlahan:", kiri+kanan)  
}
```

SS OUTPUT:



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
Masukkan bilangan bulat positif (>10): 15  
Bilangan 1: 1  
Bilangan 2: 5  
Hasil penjumlahan: 6  
● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
Masukkan bilangan bulat positif (>10): 12345  
Bilangan 1: 12  
Bilangan 2: 345  
Hasil penjumlahan: 357  
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> |
```

DESKRIPSI:

Program Go di atas digunakan untuk memisahkan sebuah bilangan bulat positif (lebih dari 10) menjadi dua bagian berdasarkan jumlah digitnya, lalu menjumlahkan kedua bagian tersebut. Pertama, program membaca input bilangan n dari pengguna, lalu menghitung panjang digit (pgj) bilangan tersebut. Kemudian, program menentukan nilai pemisah ($pisah$) berdasarkan setengah panjang digit (atau lebih satu digit jika panjangnya ganjil). Bilangan dibagi menjadi dua: bagian kiri ($kiri$) didapat dari hasil pembagian n dengan $pisah$, dan bagian kanan ($kanan$) adalah sisa bagi dari pembagian tersebut. Akhirnya, kedua bilangan tersebut ditampilkan bersama hasil penjumlahannya. Program ini berguna, misalnya, untuk memisahkan bilangan menjadi dua bagian simetris dari kiri dan kanan.

KODE PROGRAM NO2(BUKBER IF):

```
package main

import "fmt"

func main() {

    var jumlahPeserta_103112400077, nomorKartu, jumlahA, jumlahB,
    jumlahC int

    fmt.Print("Masukkan jumlah peserta: ")

    fmt.Scan(&jumlahPeserta_103112400077)

    for i := 1; i <= jumlahPeserta_103112400077; i++ {

        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i)

        fmt.Scan(&nomorKartu)

        semuaSama, semuaBerbeda := true, true

        angkaTerakhir := nomorKartu % 10

        angkaYangSudahDilihat := 0

        for sementara := nomorKartu; sementara > 0; sementara /=
10 {

            angka := sementara % 10

            if angka != angkaTerakhir {

                semuaSama = false

            }

            angkaDitemukan := false

            for pengecekan := angkaYangSudahDilihat; pengecekan >
0; pengecekan /= 10 {

                if pengecekan%10 == angka {

                    angkaDitemukan = true

                    break
```

```

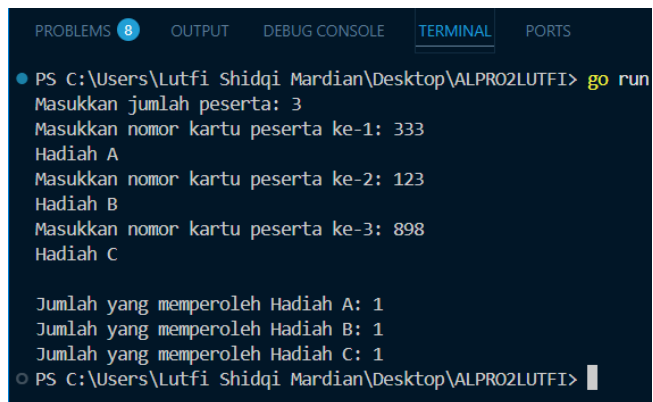
        }
    }
    if angkaDitemukan {
        semuaBerbeda = false
    }
    angkaYangSudahDilihat = angkaYangSudahDilihat*10 +
angka
    }
    if semuaSama {
        fmt.Println("Hadiah A")
        jumlahA++
    } else if semuaBerbeda {
        fmt.Println("Hadiah B")
        jumlahB++
    } else {
        fmt.Println("Hadiah C")
        jumlahC++
    }
}

fmt.Printf("\nJumlah yang memperoleh Hadiah A: %d\n",
jumlahA)

fmt.Printf("Jumlah yang memperoleh Hadiah B: %d\n", jumlahB)
fmt.Printf("Jumlah yang memperoleh Hadiah C: %d\n", jumlahC)
}

```

SS OUTPUT:

A screenshot of a terminal window with a dark blue background. At the top, there are tabs: 'PROBLEMS' (with a blue circle containing the number 8), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and highlighted in blue), and 'PORTS'. The terminal shows the following text:

```
PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run  
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 C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI>
```

DESKRIPSI:

Program ini digunakan untuk menentukan jenis hadiah yang diperoleh setiap peserta berdasarkan nomor kartu yang dimasukkan. Setiap peserta diminta memasukkan nomor kartu, lalu program akan memeriksa pola digit di dalamnya. Jika seluruh digit pada nomor kartu sama, peserta akan mendapatkan Hadiah A. Jika semua digit berbeda satu sama lain, peserta akan mendapatkan Hadiah B. Namun, jika terdapat campuran antara digit yang sama dan berbeda, maka peserta akan mendapatkan Hadiah C. Selama proses, program juga menghitung total jumlah peserta yang memperoleh masing-masing hadiah, kemudian menampilkannya di akhir program.

KODE PROGRAM NO3(PERKALIAN PENJUMLAHAN):

```
package main

import "fmt"

func dikali(n, m, hasil int) int {

    if m == 0 {

        return hasil

    }

    return dikali(n, m-1, hasil+n)

}

func main() {

    var n_103112400077, m int

    fmt.Print("Masukkan bilangan n: ")

    fmt.Scan(&n_103112400077)

    fmt.Print("Masukkan bilangan m: ")

    fmt.Scan(&m)

    if m < 0 {

        m = -m

        n_103112400077 = -n_103112400077

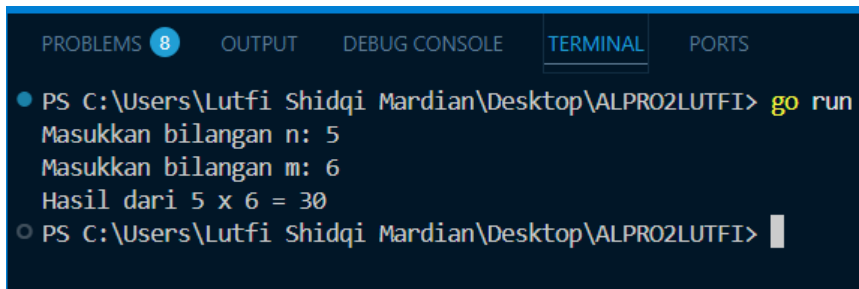
    }

    hasil := dikali(n_103112400077, m, 0)

    fmt.Printf("Hasil dari %d x %d = %d", n_103112400077, m, hasil)

}
```

SS OUTPUT:



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> go run
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Hasil dari 5 x 6 = 30
○ PS C:\Users\Lutfi Shidqi Mardian\Desktop\ALPRO2LUTFI> |
```

DESKRIPSI:

Program Go di atas digunakan untuk melakukan perkalian dua bilangan bulat menggunakan rekursi tanpa menggunakan operator perkalian (*). Fungsi dikali menerima tiga parameter: bilangan n, bilangan m, dan hasil sebagai akumulator. Jika m sama dengan 0, maka hasil akhir dikembalikan; jika tidak, fungsi akan memanggil dirinya sendiri dengan m-1 dan menambahkan n ke hasil. Di fungsi main, program menerima input n dan m dari pengguna. Jika m negatif, nilainya diubah menjadi positif, dan tanda n juga dibalik agar hasil perkalian tetap valid secara matematis. Program kemudian mencetak hasil perkalian $n \times m$ berdasarkan proses rekursif tersebut.