

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 12
WHILE-LOOP



Disusun oleh:

Akhmad Noval Annur

109082500100

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

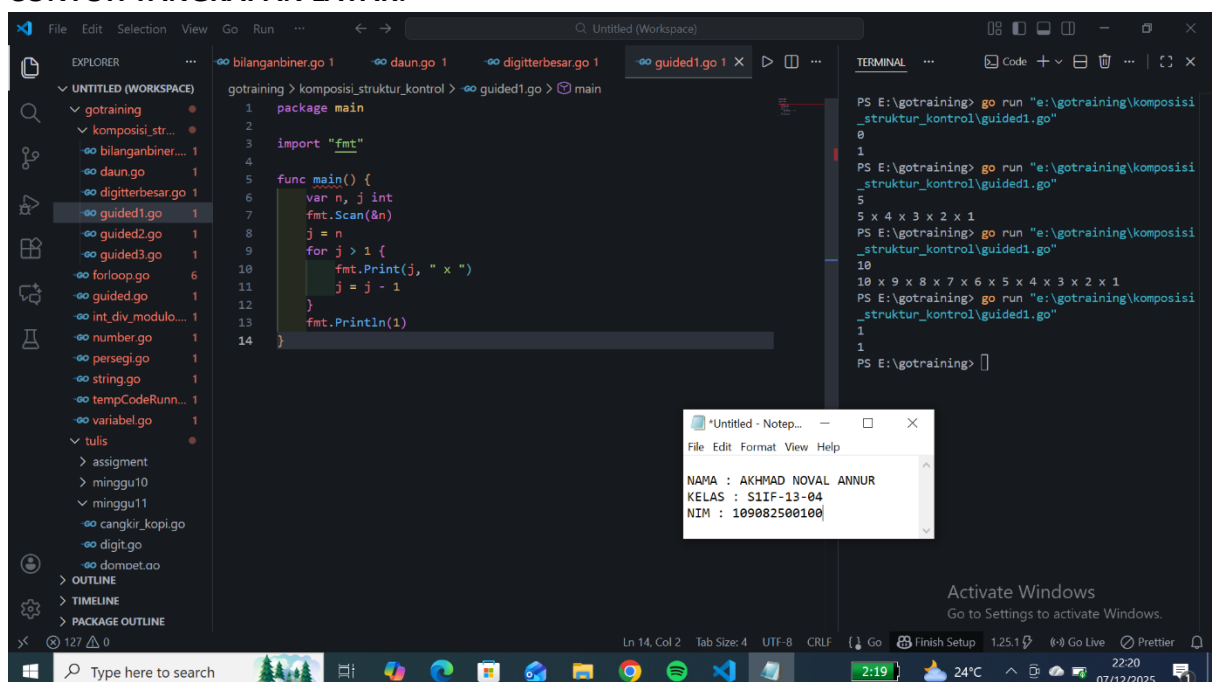
import "fmt"

func main() {
    var n, j int
    fmt.Scan(&n)
    j = n
    for j > 1 {
        fmt.Print(j, " x ")
        j = j - 1
    }
    fmt.Println(1)
}
```

Screenshoot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)

CONTOH TANGKAPAN LAYAR:



Deskripsi program

Program ini digunakan untuk menampilkan susunan faktorial dari sebuah bilangan yang dimasukkan oleh pengguna. Ketika pengguna memasukkan nilai n , program tidak menghitung hasil faktorialnya, tetapi hanya menampilkan bentuk deret dari proses faktorial tersebut. Program mulai dengan menyalin nilai n ke variabel lain bernama j , yang digunakan sebagai penghitung mundur. Dengan begitu, nilai asli n tetap terjaga sementara j dapat berkurang setiap iterasi.

Selama nilai j masih lebih dari satu, program akan mencetak angka tersebut diikuti tanda “ x ” untuk menunjukkan bahwa proses faktorial masih berlanjut. Setiap kali angka dicetak, nilai j dikurangi satu sehingga deret berjalan mundur. Ketika j sudah tidak lebih besar dari satu, perulangan berhenti dan program mencetak angka 1 sebagai penutup deret. Hasil akhirnya adalah rangkaian lengkap pembentuk faktorial, misalnya “ $5 \times 4 \times 3 \times 2 \times 1$ ”, yang menggambarkan urutan perkalian faktorial secara utuh.

2. Guided 2

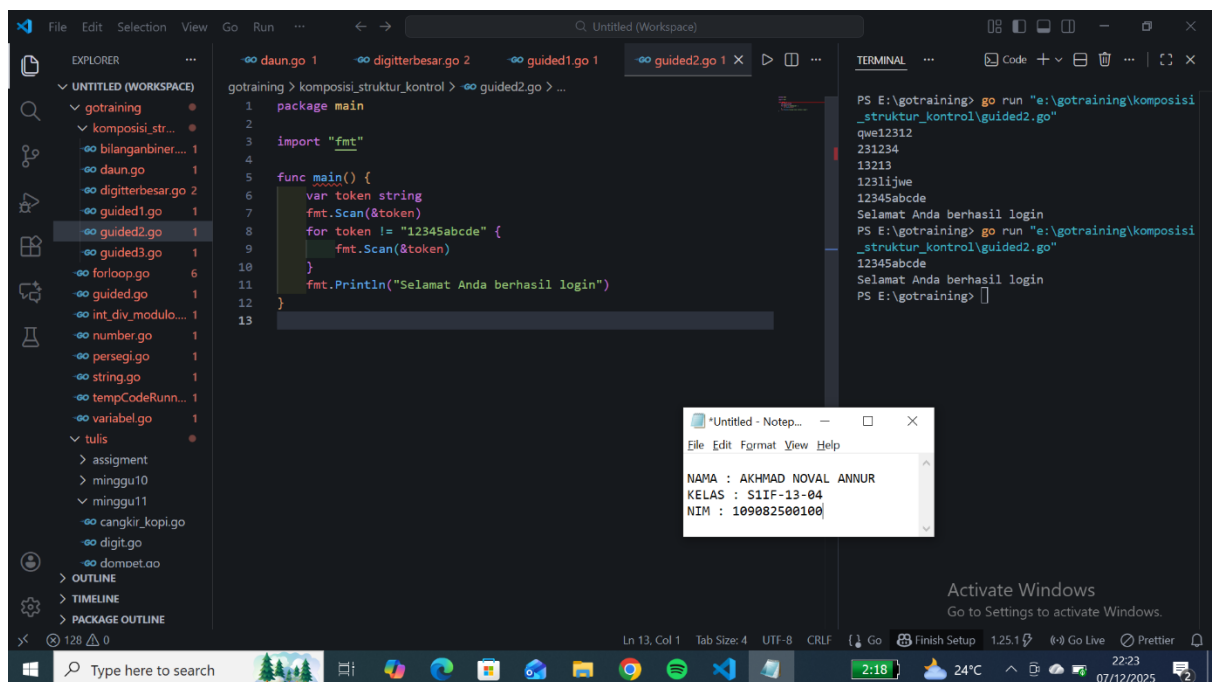
Source Code

```
package main

import "fmt"

func main() {
    var token string
    fmt.Scan(&token)
    for token != "12345abcde" {
        fmt.Scan(&token)
    }
    fmt.Println("Selamat Anda berhasil login")
}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk mengecek apakah token yang dimasukkan pengguna sesuai dengan token yang dianggap valid, yaitu "12345abcde". Saat program dijalankan, pengguna langsung diminta memasukkan satu string sebagai token pertama.

Token tersebut segera dibandingkan dengan token yang valid untuk memastikan apakah pengguna sudah memasukkan data yang benar.

Jika token pertama masih salah, program masuk ke perulangan yang terus meminta token baru sampai pengguna memasukkan token yang cocok. Selama token belum benar, program tidak memberitahukan jumlah kesalahan atau pesan tambahan lain—ia hanya meminta input berulang. Ketika pengguna akhirnya memasukkan token valid, perulangan berhenti dan program menampilkan pesan “Selamat Anda berhasil login”, yang menandakan bahwa proses autentikasi telah berhasil dilakukan. Dengan cara ini, program mensimulasikan cara kerja login paling dasar yang hanya mengizinkan pengguna masuk jika informasi yang diberikan benar.

3. Guided 3

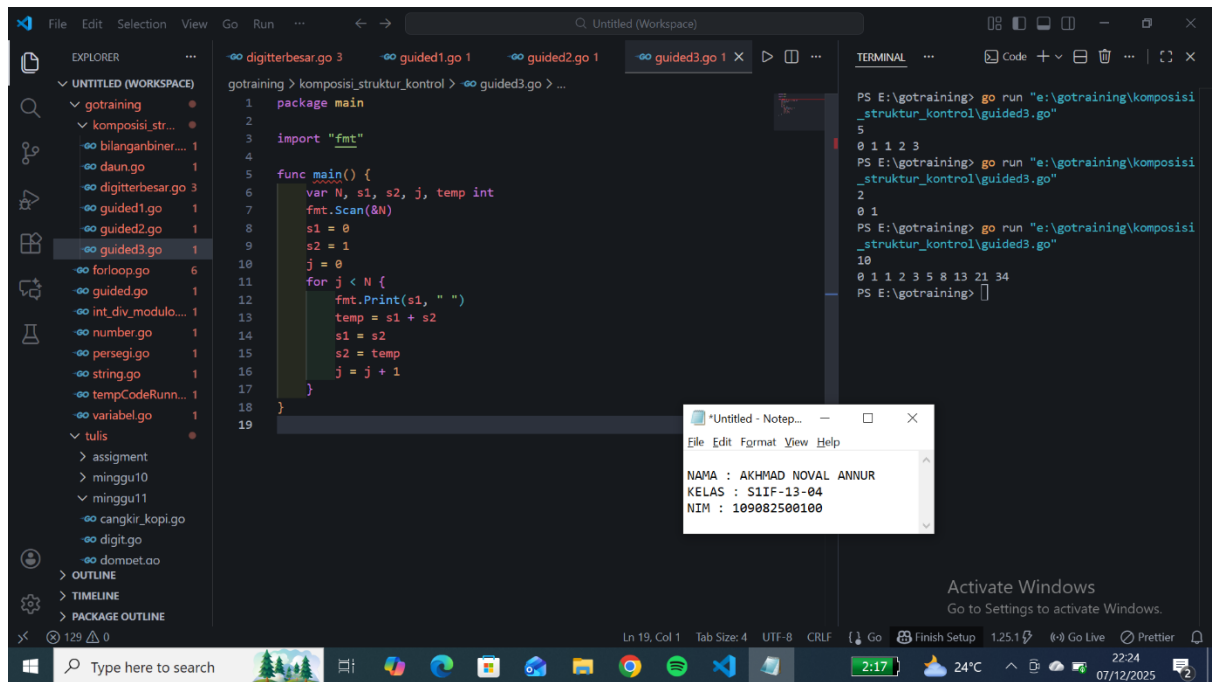
Source Code

```
package main

import "fmt"

func main() {
    var N, s1, s2, j, temp int
    fmt.Scan(&N)
    s1 = 0
    s2 = 1
    j = 0
    for j < N {
        fmt.Print(s1, " ")
        temp = s1 + s2
        s1 = s2
        s2 = temp
        j = j + 1
    }
}
```

Screenshoot program



Deskripsi program

Program ini dibuat untuk menampilkan sejumlah bilangan pertama dalam deret Fibonacci berdasarkan input dari pengguna. Deret Fibonacci sendiri dimulai dari angka 0 dan 1, kemudian setiap angka berikutnya merupakan hasil penjumlahan dua angka sebelumnya. Untuk menghasilkan deret ini, program menyiapkan dua variabel awal, yaitu $s1 = 0$ dan $s2 = 1$, yang mewakili dua nilai pertama Fibonacci.

Ketika pengguna memasukkan nilai N , program akan mencetak N angka Fibonacci secara berurutan. Pada setiap perulangan, nilai $s1$ dicetak karena ia menyimpan angka Fibonacci saat ini. Selanjutnya program menghitung angka berikutnya dengan cara menjumlahkan $s1$ dan $s2$, lalu memperbarui nilai kedua variabel tersebut agar deret dapat terus berlanjut. Perulangan hanya berjalan sampai jumlah angka yang dicetak mencapai N . Dengan proses ini, program mampu menghasilkan deret Fibonacci dengan struktur yang benar dan sesuai jumlah yang diminta pengguna.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var user, pw string
    gagal := 0

    fmt.Scan(&user, &pw)

    for user != "Admin" || pw != "Admin" {
        gagal = gagal + 1
        fmt.Scan(&user, &pw)
    }

    fmt.Printf("%d percobaan gagal login\n", gagal)
}
```

Screenshoot program

2. Tugas 2

Source code

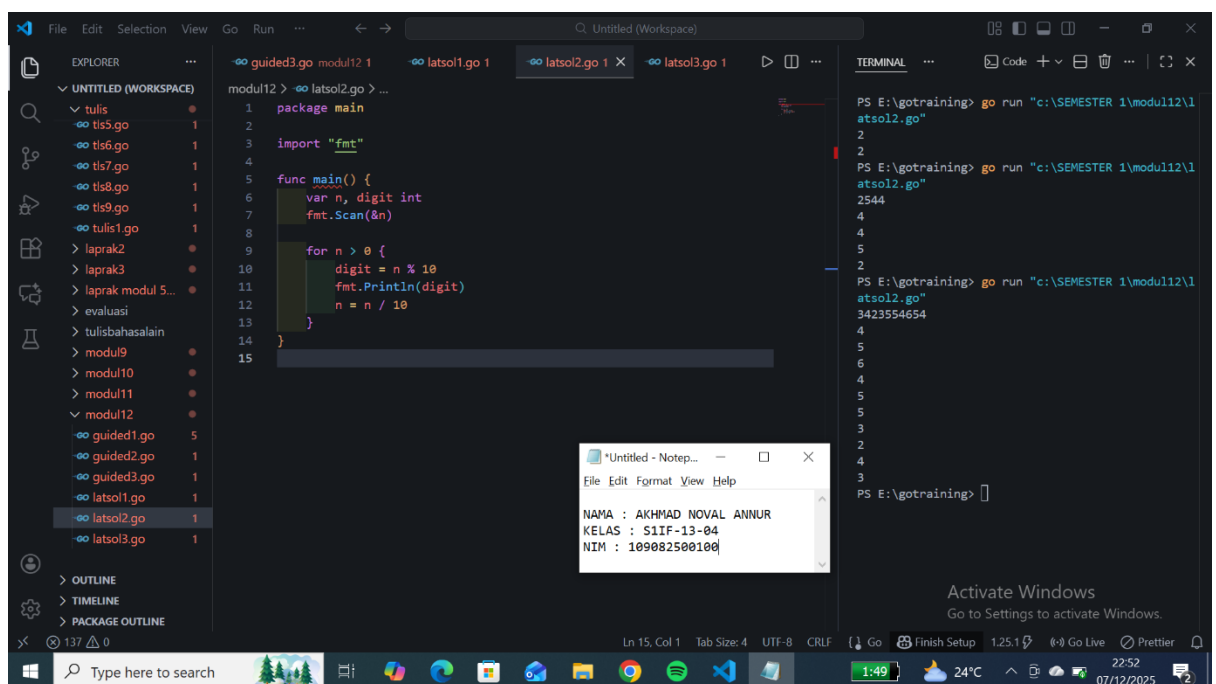
```
package main

import "fmt"

func main() {
    var n, digit int
    fmt.Scan(&n)

    for n > 0 {
        digit = n % 10
        fmt.Println(digit)
        n = n / 10
    }
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk memecah sebuah bilangan bulat positif menjadi digit-digit penyusunnya dan menampilkannya mulai dari digit paling belakang. Ketika pengguna memasukkan sebuah angka, program mengambil digit terakhir dengan menggunakan operasi sisa bagi ($n \% 10$). Digit ini langsung ditampilkan sebagai hasil. Setelah itu, digit yang sudah diambil dihapus dari bilangan dengan menjalankan operasi $n = n / 10$.

Program mengulangi proses tersebut selama nilai n masih lebih dari nol. Dengan begitu, setiap iterasi menampilkan satu digit dari arah kanan ke kiri. Cara kerja ini membuat program dapat menampilkan seluruh digit bilangan satu per satu tanpa menggunakan array atau penyimpanan tambahan. Hasil akhirnya adalah urutan digit yang ditampilkan secara terbalik dibandingkan bentuk aslinya, misalnya input 2544 akan menghasilkan 4, 4, 5, 2 secara berurutan.

3. Tugas 3

Source code

```
package main

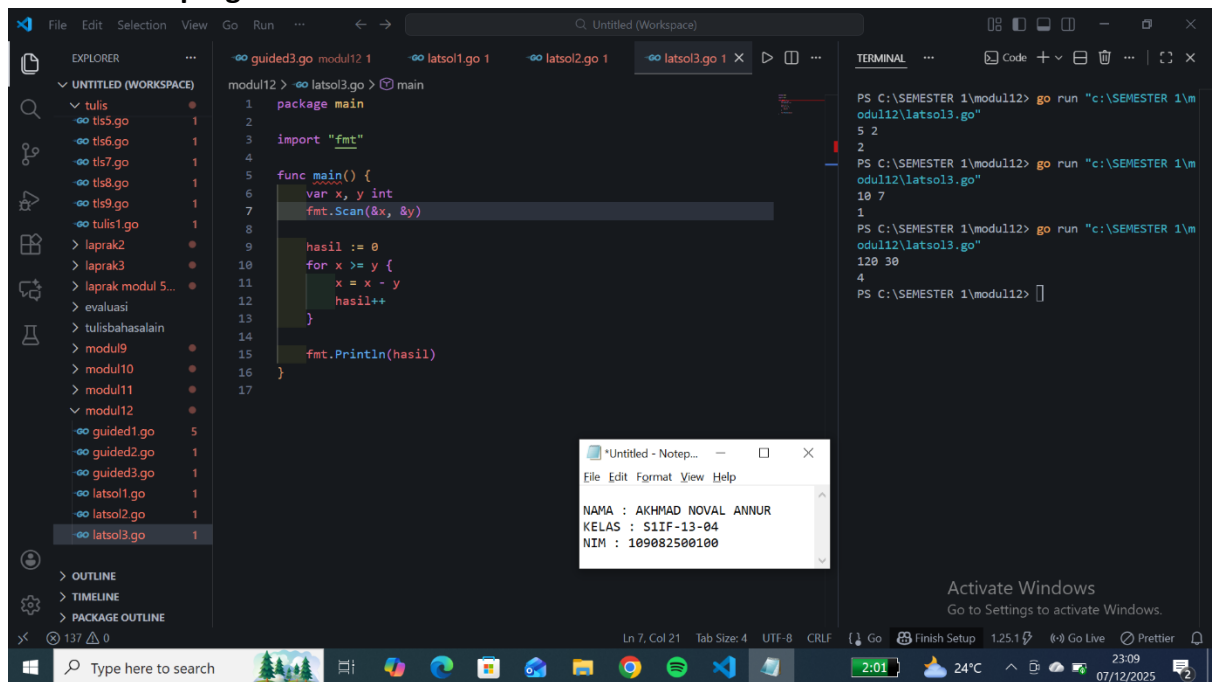
import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x, &y)

    hasil := 0
    for x >= y {
        x = x - y
        hasil++
    }

    fmt.Println(hasil)
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menghitung pembagian bilangan bulat x terhadap y ($x \div y$) tanpa menggunakan operator pembagian. Sebagai gantinya, program menggunakan metode pengurangan berulang. Setelah pengguna memasukkan kedua angka, yaitu x sebagai bilangan yang akan dibagi dan y sebagai pembaginya, program menyiapkan variabel `hasil` untuk menghitung berapa kali y bisa dikurangkan dari x .

Program kemudian menjalankan perulangan selama x masih lebih besar atau sama dengan y . Pada setiap iterasi, nilai x dikurangi y , dan variabel hasil ditambah satu. Proses ini sebenarnya meniru konsep pembagian manual: “berapa kali angka y bisa diambil dari x ”. Ketika x sudah tidak dapat dikurangi lagi karena nilainya lebih kecil dari y , perulangan berhenti dan nilai hasil ditampilkan sebagai hasil pembagian bulat. Dengan cara ini, program dapat melakukan pembagian tanpa menggunakan simbol “/” namun tetap menghasilkan nilai yang benar.