

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)
}
```

Screenshot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)
CONTOH TANGKAPAN LAYAR:

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The Explorer sidebar on the left lists several Go files in a directory structure under 'gotraining'. The 'guided1.go' file is currently open in the editor, displaying the source code provided above. To the right of the editor is a 'TERMINAL' tab showing command-line output. The terminal window shows the execution of the Go code, with the output being the multiplication of numbers from 10 down to 1, followed by a final '1'. Below the terminal is a small 'Untitled - Notepad' window containing student information: NAMA : AKHMAD NOVAL ANNUR, KELAS : S1IF-13-04, and NIM : 109082500100. At the bottom of the screen is the Windows taskbar, which includes icons for File Explorer, Task View, Edge browser, File Explorer, and others, along with system status indicators like battery level and network connection.

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")
}
```

Screenshot program

The screenshot shows a Go development environment with the following components:

- EXPLORER**: Shows the workspace structure with files like `daun.go`, `digitterbesar.go`, `guided1.go`, and `guided2.go`.
- TERMINAL**: Displays command-line output from running the program. It shows the user entering tokens and receiving responses: "231234", "13213", "1231ijwe", and "12345abcde". It also shows the message "Selamat Anda berhasil login" (Congratulations, you have successfully logged in) twice.
- Code Editor**: Shows the source code of `guided2.go`. The code defines a `main` function that reads tokens from standard input until it finds one equal to "12345abcde". When found, it prints the message "Selamat Anda berhasil login".
- Output Window**: A small window titled "Untitled - Notepad" showing the user's input and the program's response: "NAMA : AKHMAD NOVAL ANNUR", "KELAS : S1IF-13-04", and "NIM : 109082500100".

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.Println(s1, " ")
        temp = s1 + s2
        s1 = s2
        s2 = temp
        j = j + 1
    }
}
```

Screenshoot program

```
digiterbesar.go 3  guided1.go 1  guided2.go 1  guided3.go 1
go run "e:\gotraining\komposisi_structur_kontrol\guided3.go"
5
0 1 1 2 3
PS E:\gotraining> go run "e:\gotraining\komposisi_structur_kontrol\guided3.go"
2
0 1
PS E:\gotraining> go run "e:\gotraining\komposisi_structur_kontrol\guided3.go"
10
0 1 1 2 3 5 8 13 21 34
PS E:\gotraining>
```

*Untitled - Notep... —

File Edit Format View Help

NAMA : AKHMAD NOVAL ANNUR
KELAS : SIIF-13-04
NIM : 109082500100

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)
}
```

Screenshot program

The screenshot shows a Windows desktop environment. In the center is a terminal window titled "Untitled (Workspace)" with the command "PS E:\gotraining> go run "c:\SEMESTER 1\modul12\1\atsoll1.go"" and its output:

```
User:23 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
PS E:\gotraining> go run "c:\SEMESTER 1\modul12\1\atsoll1.go"
Admin Admin
0 percobaan gagal login
PS E:\gotraining>
```

Below the terminal is a code editor window for "guided3.go" in the "modul12" workspace. The code is as follows:

```
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)
}
```

To the right of the code editor is a small "Untitled - Notepad" window containing student information:

```
NAMA : AKHMAD NOVAL ANNUR
KELAS : S1IF-13-04
NIM : 109082500100
```

Deskripsi program

Program ini bertugas menghitung berapa kali pengguna memasukkan username dan password yang salah sebelum akhirnya berhasil login. Data login yang benar sangat sederhana, yaitu username “Admin” dan password “Admin”. Saat program pertama kali dijalankan, pengguna diminta memasukkan dua nilai tersebut. Jika input sudah benar, program langsung berhenti dan menampilkan bahwa tidak ada percobaan gagal.

Namun jika input pertama salah, program masuk ke perulangan di mana ia terus meminta pengguna memasukkan ulang username dan password. Setiap kali pengguna memasukkan kombinasi yang salah, variabel gagal akan bertambah satu. Perulangan ini berhenti ketika pengguna akhirnya memasukkan data login yang benar. Setelah itu, program menampilkan total jumlah kesalahan yang telah dilakukan. Dengan cara ini, program tidak hanya memvalidasi login, tetapi juga mencatat jumlah percobaan gagal seperti sistem login sederhana pada aplikasi nyata.

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
    }
}
```

Screenshot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The left sidebar displays a file tree under 'UNTITLED (WORKSPACE)' containing several Go files: guided3.go, latsol1.go, latsol2.go, and latsol3.go. The main editor area shows the source code for latsol2.go. The terminal tab at the top right shows the command 'go run "c:\SEMESTER 1\modul12\latsol2.go"' followed by the output of the program, which prints the digits of a number. A small 'Untitled - Notepad' window is also visible, showing some student information. The taskbar at the bottom includes icons for File Explorer, Task View, Start, and various system status indicators like battery level and network.

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

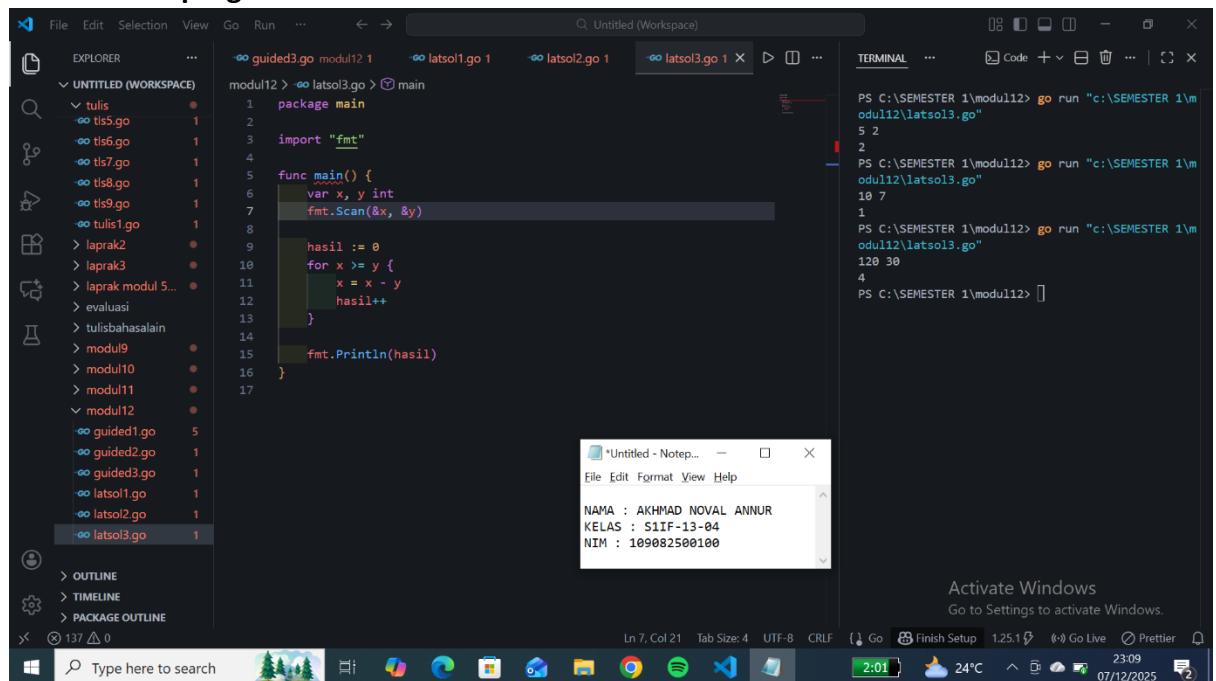
import "fmt"

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

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

    fmt.Println(hasil)
}
```

Screenshot program



Deskripsi program

Program ini digunakan untuk menghitung pembagian bilangan bulat x terhadap y ($x \text{ div } y$) 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.