

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 9
FOOR-LOP**



Disusun oleh:
MUHAMMAD NAUFAL AKMAL PEABOWO
109082500186
S1IF-13-02

Asisten Praktikum
Adithana dharma putra
Alfin Ilham Berlianto

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

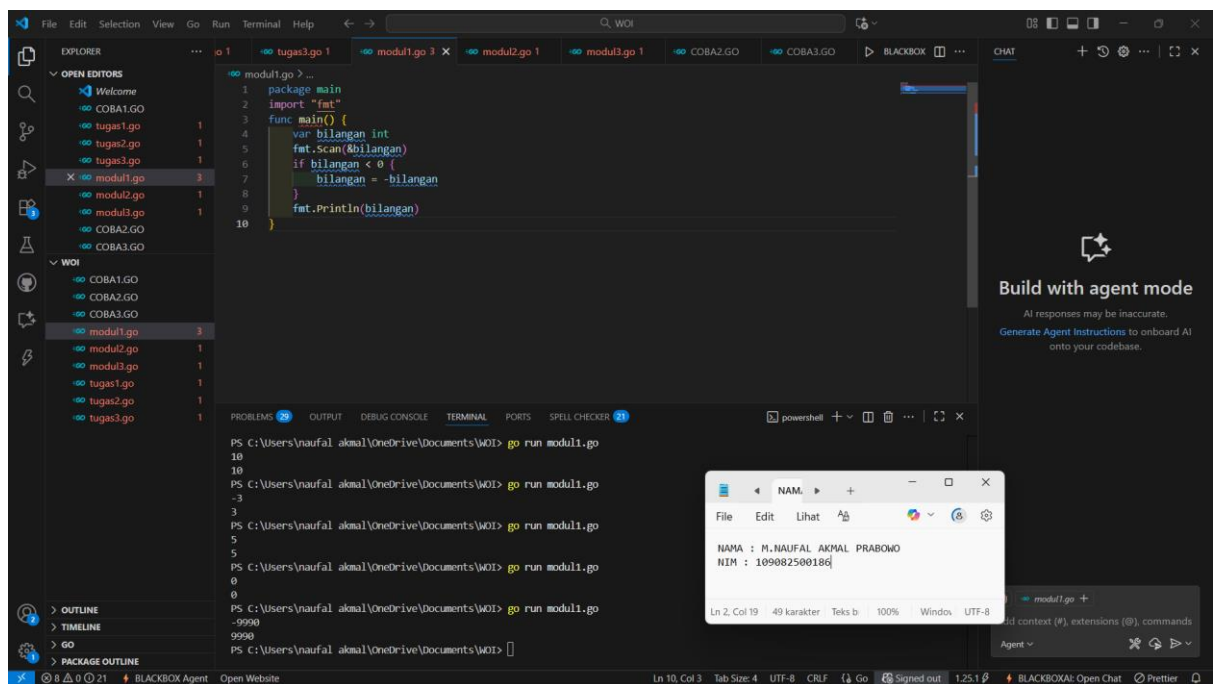
LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main
import "fmt"
func main() {
    var bilangan int
    fmt.Scan(&bilangan)
    if bilangan < 0 {
        bilangan = -bilangan
    }
    fmt.Println(bilangan)
}
```

Screenshoot program



Deskripsi program

Gambar tersebut menampilkan lingkungan pengembangan terintegrasi (IDE) Visual Studio Code yang menampilkan kode sumber Go (Golang) dalam berkas bernama modul1.go. Kode tersebut mendefinisikan fungsi main yang membaca input bilangan bulat dari pengguna menggunakan `fmt.Scan` ke dalam variabel `bilangan`, dan kemudian menggunakan pernyataan `if` untuk memeriksa apakah bilangan kurang dari nol; jika ya, nilai bilangan dikalikan dengan `-1` (mengubahnya menjadi positif), sebelum akhirnya nilai akhir dicetak menggunakan `fmt.Println`. Jendela terminal di bagian bawah menunjukkan riwayat eksekusi kode (menggunakan perintah `go run`

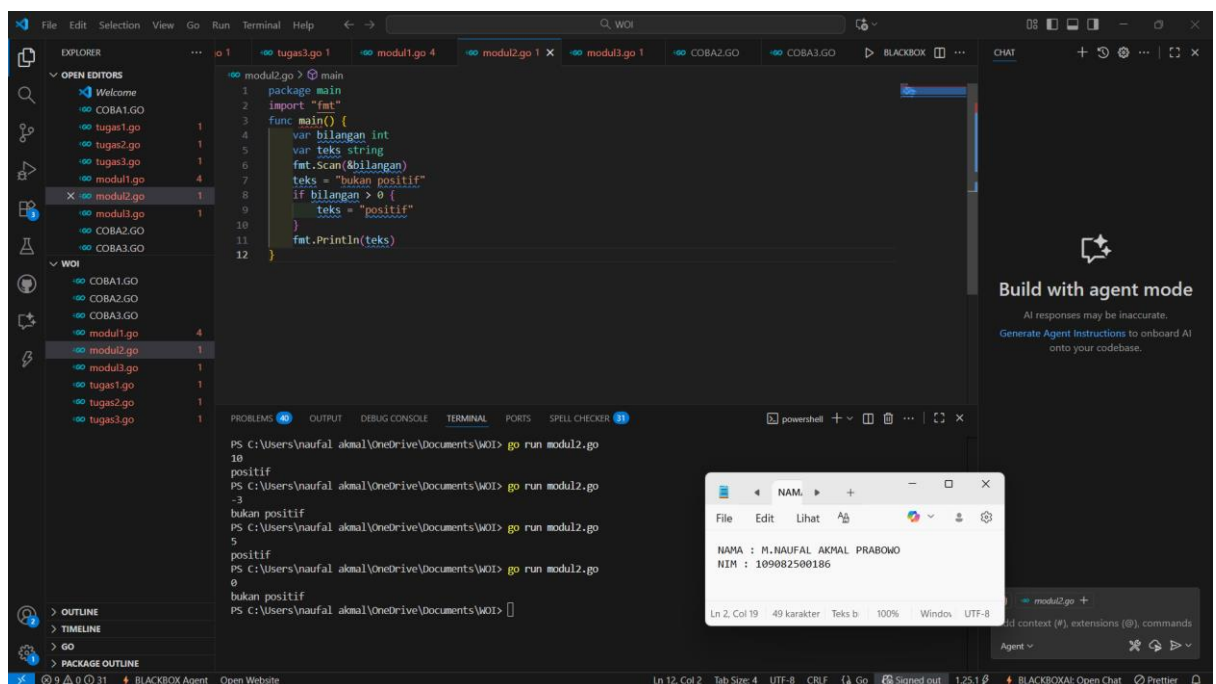
modul1.go) dengan beberapa input dan output, yang mengilustrasikan fungsi program untuk mengubah input negatif menjadi positif.

2. Guided 2

Source Code

```
package main
import "fmt"
func main() {
    var bilangan int
    var teks string
    fmt.Scan(&bilangan)
    teks = "bukan positif"
    if bilangan > 0 {
        teks = "positif"
    }
    fmt.Println(teks)
}
```

Screenshoot program



Deskripsi program

Gambar ini menampilkan lingkungan pengembangan Visual Studio Code dengan fokus pada berkas Go (Golang) bernama modul2.go. Kode di dalamnya menunjukkan implementasi fungsi main yang membaca input bilangan bulat ke dalam variabel bilangan dan menginisialisasi variabel teks sebagai string "bukan positif". Sebuah pernyataan if kemudian memeriksa apakah bilangan bernilai lebih besar atau sama dengan nol (bilangan >= 0); jika kondisi ini benar, variabel teks diperbarui menjadi "positif", dan nilai akhir dari variabel teks kemudian dicetak ke konsol

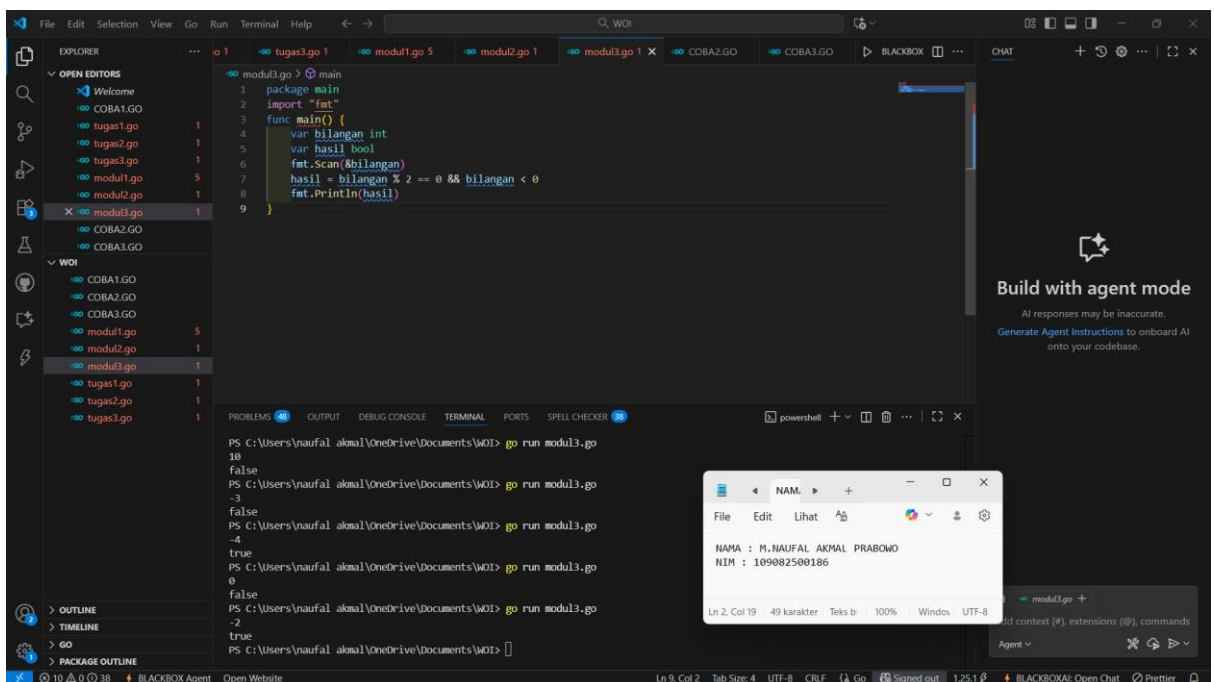
menggunakan `fmt.Println`. Jendela terminal di bagian bawah menunjukkan riwayat eksekusi program `modul2.go`, yang secara efektif menguji input positif (misalnya, 10 menghasilkan "positif") dan input negatif (misalnya, -3 menghasilkan "bukan positif").

3. Guided 3

Source Code

```
package main
import "fmt"
func main() {
    var bilangan int
    var hasil bool
    fmt.Scan(&bilangan)
    hasil = bilangan % 2 == 0 && bilangan < 0
    fmt.Println(hasil)
}
```

Screenshoot program



Deskripsi program

Gambar tersebut menampilkan Visual Studio Code yang menampilkan kode sumber Go (Golang) dalam berkas bernama `modul3.go`, yang dirancang untuk melakukan operasi logika boolean. Fungsi `main` membaca input bilangan bulat ke dalam variabel `bilangan` dan kemudian menetapkan variabel `hasil` dengan hasil dari ekspresi boolean: `bilangan >= 2 && bilangan < 0`. Ekspresi ini menggunakan operator AND (&&), yang secara logika selalu bernilai `false` karena sebuah bilangan tidak mungkin secara bersamaan lebih besar atau sama dengan dua dan kurang dari nol. Jendela terminal

di bagian bawah menunjukkan hasil eksekusi program (`go run modul3.go`) dengan berbagai input (seperti 10, -3, 0, dan -2), di mana semua output yang dicetak ke konsol adalah false, sesuai dengan logika ekspresi tersebut. Selain itu, seperti pada gambar sebelumnya,

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

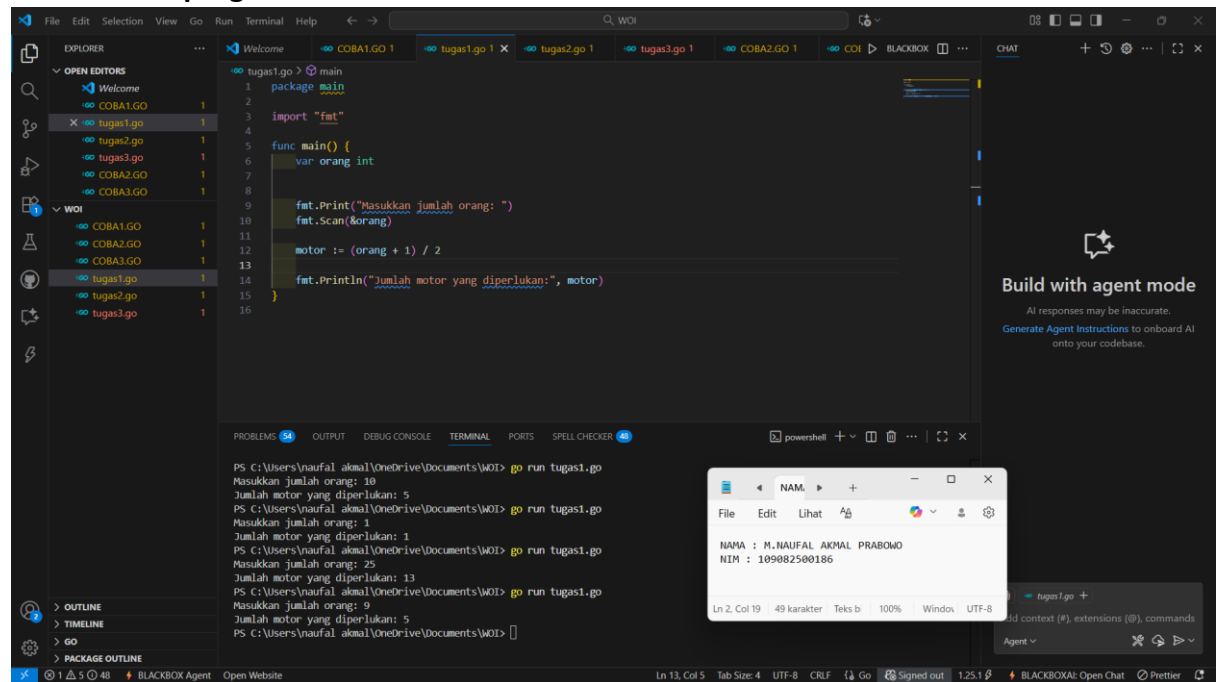
func main() {
    var orang int

    fmt.Print("Masukkan jumlah orang: ")
    fmt.Scan(&orang)

    motor := (orang + 1) / 2

    fmt.Println("Jumlah motor yang diperlukan:", motor)
}
```

Screenshoot program



Deskripsi program

Gambar ini memperlihatkan antarmuka Visual Studio Code yang menampilkan kode Go dalam berkas `tugas1.go`, yang berfungsi sebagai kalkulator kebutuhan motor.

Kode tersebut mendefinisikan fungsi main yang meminta pengguna memasukkan jumlah orang melalui `fmt.Print` dan kemudian membaca input bilangan bulat ke dalam variabel `orang`. Program kemudian menghitung jumlah motor yang diperlukan menggunakan rumus $\text{motor} = (\text{orang} + 1) / 2$. Logika ini menunjukkan bahwa setiap motor dapat mengangkut maksimal dua orang, dan ditambahkan satu untuk pembulatan ke atas (*ceiling*) sebelum pembagian integer agar hasilnya selalu dibulatkan ke atas untuk mengakomodasi sisa orang. Jendela terminal di bagian bawah menunjukkan riwayat eksekusi program, mengilustrasikan hasil perhitungan: input 10 orang memerlukan 5 motor, 25 orang memerlukan 13 motor, dan 9 orang memerlukan 5 motor. Seperti gambar-gambar sebelumnya,

2. Tugas 2

Source code

```
package main

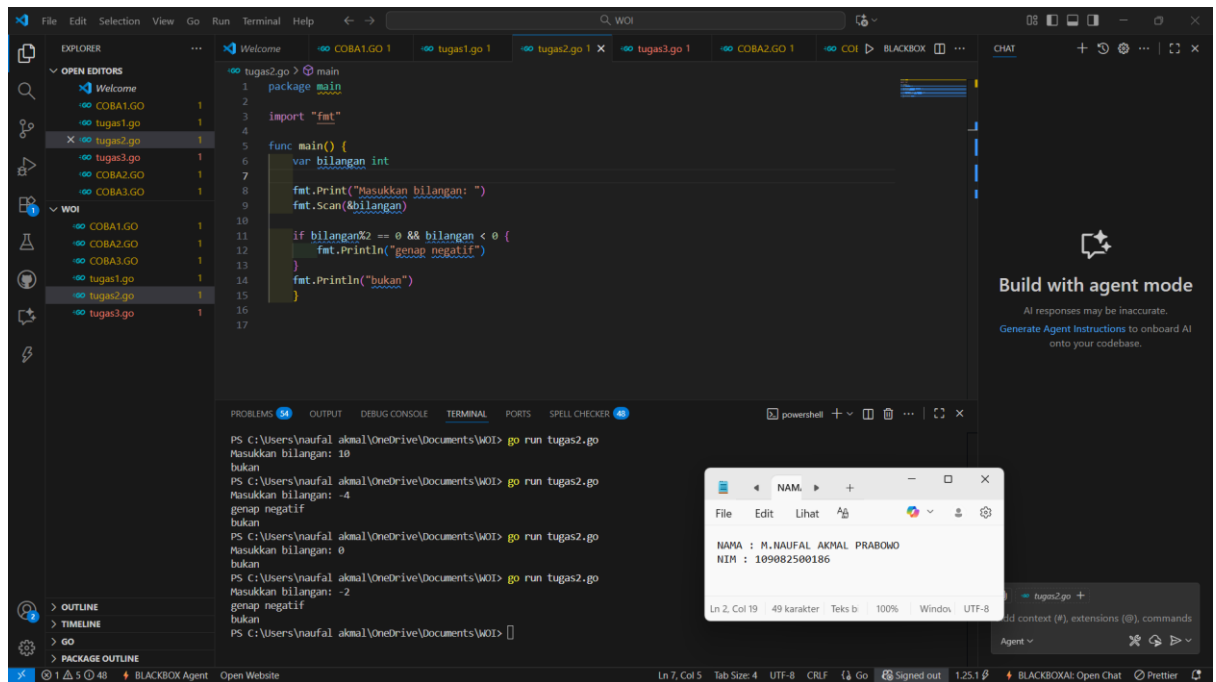
import "fmt"

func main() {
    var bilangan int

    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&bilangan)

    if bilangan%2 == 0 && bilangan < 0 {
        fmt.Println("genap negatif")
    }
    fmt.Println("bukan")
}
```

Screenshoot program



Deskripsi program

Gambar ini menampilkan lingkungan Visual Studio Code yang fokus pada berkas Go bernama `tugas2.go`, yang berisi fungsi `main` yang dirancang untuk menguji properti bilangan bulat. Program meminta input bilangan bulat dari pengguna ke dalam variabel `bilangan`. Sebuah pernyataan `if` bersarang (*nested if*) mengecek dua kondisi: pertama, apakah bilangan habis dibagi dua (`bilangan % 2 == 0`, yang berarti genap), dan kedua, apakah bilangan kurang dari nol (`bilangan < 0`). Jika kedua kondisi ini terpenuhi (yaitu, bilangan tersebut genap dan negatif), program mencetak "genap negatif". Terlepas dari hasil pemeriksaan `if`, program secara default mencetak "bukan" setelah blok `if`. Jendela terminal di bagian bawah menunjukkan hasil eksekusi: input `-4` dan `-2` menghasilkan "genap negatif" diikuti oleh "bukan", sementara input `10` dan `0` hanya menghasilkan "bukan", menunjukkan bahwa hanya bilangan bulat genap negatif yang memicu pencetakan pertama.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var x, y int

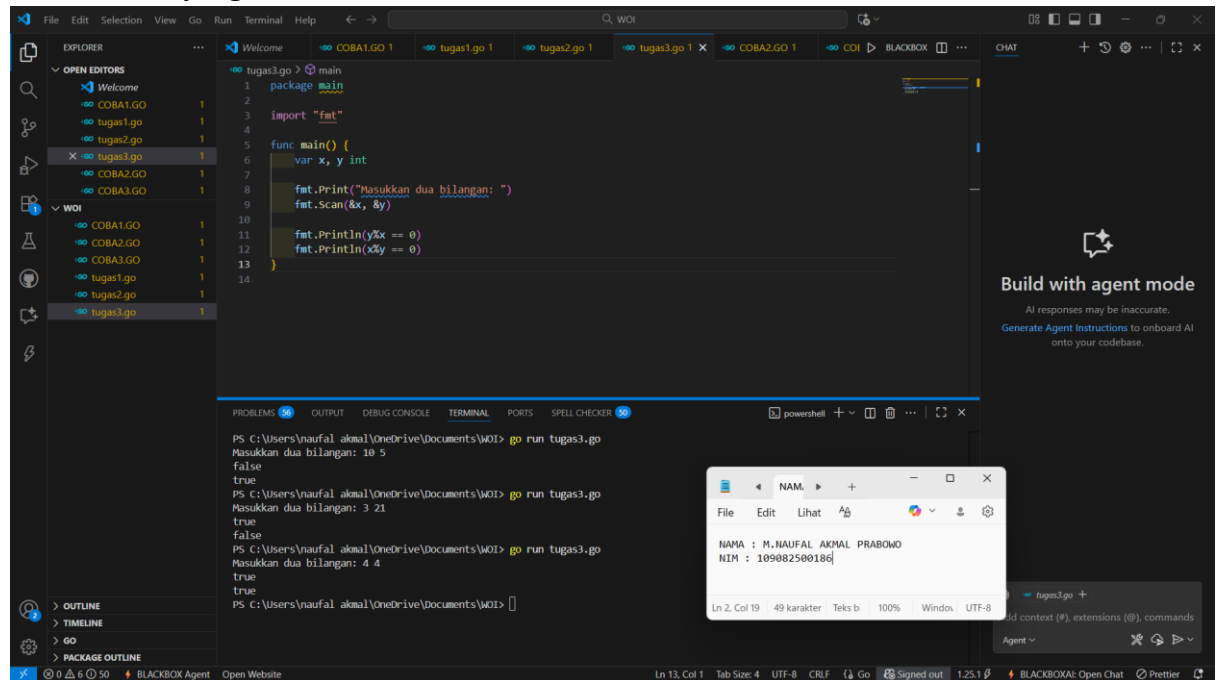
    fmt.Print("Masukkan dua bilangan: ")
    fmt.Scan(&x, &y)
```

```

    fmt.Println(y%x == 0)
    fmt.Println(x%y == 0)
}

```

Screenshoot program



Deskripsi program

Gambar ini menampilkan lingkungan Visual Studio Code yang fokus pada berkas Go bernama *tugas3.go*, yang dirancang untuk **membandingkan dua bilangan bulat**. Fungsi *main* mendeklarasikan dua variabel integer, *x* dan *y*, kemudian meminta pengguna untuk memasukkan dua bilangan melalui *fmt.Print* dan membacanya menggunakan *fmt.Scan(&x, &y)*. Program ini kemudian mencetak dua baris output boolean: baris pertama mencetak hasil dari ekspresi $x == y$, yang bernilai **true** jika *x* sama dengan *y*, dan **false** jika berbeda; baris kedua mencetak hasil dari ekspresi $x < y$, yang bernilai **true** jika *x* lebih kecil dari *y*, dan **false** sebaliknya. Jendela terminal di bagian bawah menunjukkan riwayat eksekusi program dengan beberapa pasangan input, seperti 10 5 (menghasilkan false dan false), 3 21 (menghasilkan false dan true), dan 4 4 (menghasilkan true dan false).