

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 9

IF-THEN



Disusun oleh:

Johanson Leeroy

109082500017

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 x int
    fmt.Scan(&x)

    if x < 0 {
        x = -x
    }

    fmt.Println(x)
}
```

Screenshot program

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var x int
7     fmt.Scan(&x)
8
9     if x < 0 {
10         x = -x
11     }
12     fmt.Println(x)
13 }
14
15


```

109082500017
S1IF-13-04
Johanson Leeroy

Ln 10, Col 1 187 charact Plain t 100% Wind UTF-8

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided1.go"
10
10
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided1.go"
-3
3
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided1.go"
5
5
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided1.go"
0
0
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided1.go"
-9999
9999
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9>

```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel x dengan tipe data integer/int
3. User menginputkan bilangan bulat/integer ke dalam variabel x
4. Membuat struktur kontrol if-then dengan kondisi x < 0
5. Jika kondisi bernilai true maka program akan melakukan x = -x dan Jika kondisi bernilai false maka program akan langsung mengeluarkan output berupa bilangan pada variabel x tanpa mengeksekusi program dalam struktur kontrol if-then
6. Output berupa nilai akhir dari variabel x

2. Guided 2

Source Code

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var x int
```

```

var hasil string

hasil = "Bukan Positif"

fmt.Scan(&x)

if x > 0 {

    hasil = "Positif"

}

fmt.Println(hasil)
}

```

Screenshot program

The screenshot shows a code editor with a terminal window integrated. The code editor displays the following Go code:

```

1 Guided2.go > main
2 package main
3 import "fmt"
4
5 func main() {
6     var x int
7     var hasil string
8     hasil = "Bukan Positif"
9     fmt.Scan(&x)
10
11    if x > 0 {
12        hasil = "Positif"
13    }
14    fmt.Println(hasil)
15 }

```

The terminal window shows the execution of the program and its output:

```

PS D:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9> go run "d:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9\Guided2.go"
10
Positif
PS D:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9> go run "d:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9\Guided2.go"
-3
Bukan Positif
PS D:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9> go run "d:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9\Guided2.go"
5
Positif
PS D:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9> go run "d:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9\Guided2.go"
0
Bukan Positif
PS D:\.new_journey\SMTR_1\ALGORITMA DAN PEMROGRAMAN\Minggy_9>

```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel x dengan tipe data integer/int
3. Mendeklarasikan variabel hasil dengan tipe data string
4. Memberi string “Bukan Positif” pada variabel hasil

5. User menginputkan bilangan bulat/integer ke dalam variabel x
6. Membuat struktur kontrol if-then dengan kondisi $x > 0$
7. Jika kondisi bernilai true maka program akan mengubah string pada variabel hasil dari “Bukan Positif” menjadi “Positif” dan jika kondisi bernilai false maka program akan langsung mengeluarkan output berupa string pada variabel hasil tanpa mengeksekusi program dalam struktur kontrol if-then
8. Output berupa string akhir pada variabel hasil

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var x int
    var cek bool
    fmt.Scan(&x)
    if x < 0 {
        cek = x%2 == 0
    }
    fmt.Println(cek)
}
```

Screenshot program

The screenshot shows a code editor interface with a terminal window below it. The code editor has a dark theme and displays the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x int
7     var cek bool
8     fmt.Scan(&x)
9     if x < 0 {
10         cek = x%2 == 0
11     }
12     fmt.Println(cek)
13 }
14
```

The terminal window shows the execution of the program and its output:

```
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided3.go"
10
false
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided3.go"
-3
false
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided3.go"
-4
true
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided3.go"
0
false
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Guided3.go"
-2
true
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9>
```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel x dengan tipe data integer/int
3. Mendeklarasikan variabel cek dengan tipe data boolean
4. User menginputkan bilangan bulat/integer ke dalam variabel x
5. Membuat struktur kontrol if-then dengan kondisi $x < 0$
6. Jika kondisi bernilai true maka program akan melakukan $cek = x \% 2 == 0$ yang artinya jika $x \text{ mod } 2 = 0$ maka variabel cek akan bernilai true dan Jika kondisi pada struktur kontrol if-then bernilai false maka program akan langsung mengeluarkan output berupa bilangan pada variabel x tanpa mengeksekusi program dalam struktur kontrol if-then
7. Output berupa nilai akhir dari variabel cek

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var motor, penumpang int
    fmt.Scan(&penumpang)

    if penumpang > 0 {
        motor = (penumpang + 1) / 2
    }

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

Screenshoot program

The screenshot shows a code editor interface with a terminal window below it. The terminal window displays the execution of a Go program named 'Tugas1.go'. The program calculates the number of motors needed based on the input number of passengers. The terminal output shows several runs of the program with different passenger counts (5, 13, and 9) and their corresponding motor requirements (1, 13, and 9).

```
-> Tugas1.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var motor, penumpang int
7     fmt.Scan(&penumpang)
8
9     if penumpang > 0 {
10         motor = (penumpang + 1) / 2
11     }
12     fmt.Println("Jumlah motor dibutuhkan: ", motor)
13 }
14
```

```
File Edit View Aa X
109082500017
S1F-13-04
Johnson Leeroy

Ln 10, Col 1 | 187 charact Plain t 100% Wind UTF-8
```

```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas1.go"
10
Jumlah motor dibutuhkan: 5
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas1.go"
1
Jumlah motor dibutuhkan: 1
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas1.go"
25
Jumlah motor dibutuhkan: 13
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas1.go"
9
Jumlah motor dibutuhkan: 5
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> ]
```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel motor, penumpang dengan tipe data integer/int
3. User menginputkan bilangan bulat/integer ke dalam variabel penumpang
4. Membuat struktur kontrol if-then dengan kondisi penumpang > 0
5. Jika kondisi bernilai true maka program akan melakukan motor = (pengumpang + 1) / 2 dan Jika kondisi bernilai false maka program akan langsung mengeluarkan output berupa bilangan pada variabel motor tanpa mengeksekusi program dalam struktur kontrol if-then
6. Output berupa nilai akhir dari variabel motor yaitu jumlah motor yang dibutuhkan

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
```

```

var bil int

var cek string

fmt.Scan(&bil)

cek = "bukan"

if bil < 0 && bil%2 == 0 {

    cek = "genap negatif"

}

fmt.Println(cek)
}

```

Screenshot program

The screenshot shows a code editor with a Go file named `Tugas2.go`. The code is identical to the one above. To the right of the editor is a terminal window showing the execution of the program. The terminal output is as follows:

```

PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas2.go"
10
bukan
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas2.go"
-4
genap negatif
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas2.go"
0
bukan
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas2.go"
-2
genap negatif
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9>

```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel bil dengan tipe data integer/int

3. Mendeklarasikan variabel cek dengan tipe data string
4. User menginputkan bilangan bulat/integer ke dalam variabel bil
5. Memberi string “bukan” pada variabel cek
6. Membuat struktur kontrol if-then dengan kondisi $\text{bil} < 0 \ \&\& \ \text{bil} \% 2 == 0$
7. Jika kondisi bernilai true maka program mengganti string pada variabel cek dari “bukan” menjadi “genap negatif” dan Jika kondisi bernilai false maka program akan langsung mengeluarkan output berupa string pada variabel cek tanpa mengeksekusi program dalam struktur kontrol if-then
8. Output berupa string akhir pada variabel cek

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var x, y int
    var cek bool
    fmt.Scan(&x, &y)
    cek = false
    if y%x == 0 {
        cek = true
    }
    fmt.Println(cek)
    cek = false
    if x%y == 0 {
        cek = true
    }
    fmt.Print(cek)
}
```

Screenshot program

The screenshot shows a code editor interface with a terminal window below it. The code editor displays a Go program named Tugas3.go. The terminal window shows the execution of the program and its output.

```
1go Tugas3.go > main
2    package main
3
4    import "fmt"
5
6    func main() {
7        var x, y int
8        var cek bool
9        fmt.Scan(&x, &y)
10       cek = false
11       if y%x == 0 {
12           cek = true
13       }
14       fmt.Println(cek)
15       cek = false
16       if x%y == 0 {
17           cek = true
18       }
19       fmt.Print(cek)
20   }
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas3.go"
10 5
false
true
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas3.go"
3 21
true
false
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9\Tugas3.go"
4 4
true
true
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\Minggy 9>
```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel x, y dengan tipe data integer/int
3. Mendeklarasikan variabel cek dengan tipe data boolean
4. User menginputkan bilangan bulat/integer ke dalam variabel x dan y
5. Terlebih dahulu memberikan nilai false pada variabel cek
6. Membuat struktur kontrol if-then pertama dengan kondisi $y \% x == 0$
7. Jika kondisi bernilai true maka program mengubah nilai pada variabel cek dari false menjadi true dan Jika kondisi bernilai false maka program akan langsung mengeluarkan output berupa nilai pada variabel cek tanpa mengeksekusi program dalam struktur kontrol if-then
8. Mengeluarkan output berupa nilai akhir pada variabel cek
9. Kembali memberikan nilai false pada variabel cek
10. Membuat struktur kontrol if-then kedua dengan kodisi $x \% y == 0$
11. Jika kondisi bernilai true maka program mengubah nilai pada variabel cek dari false menjadi true dan Jika kondisi bernilai false maka program akan langsung mengeluarkan output berupa nilai pada variabel cek tanpa mengeksekusi program dalam struktur kontrol if-then
12. Mengeluarkan output berupa nilai akhir pada variabel cek