

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 10**  
**TIPE DATA DAN VARIABEL**



**Disusun oleh:**

**DIVA ZAHRAH NABILA**

**109082500112**

**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## 1. Guided 1 (KTP)

### Source code

```
package main

import "fmt"

func main() {

    var usia int
    var KK bool

    fmt.Scan(&usia, &KK)

    if usia >= 17 && KK {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}
```

### Screenshot program

The screenshot shows a code editor interface with a Go file named `guided1.go` open. The code defines a `main` function that reads age and KK status from standard input, then prints whether KTP can be made based on the conditions. Below the editor is a terminal window showing the execution of the program and its output.

```
package main

import "fmt"

func main() {
    var usia int
    var KK bool

    fmt.Scan(&usia, &KK)

    if usia >= 17 && KK {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}
```

```
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided1.go"
17 true
bisa membuat KTP
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided1.go"
20 false
belum bisa membuat KTP
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided1.go"
15 true
belum bisa membuat KTP
PS D:\LAPRAK\WEEK 10>
```

### Deskripsi program

Program ini dibuat untuk mengecek apakah seseorang bisa membuat KTP atau belum bisa membuat KTP, berdasarkan usia dan kepemilikan Kartu Keluarga (KK).

1. **package main** Menandakan bahwa program ini adalah program utama (bukan package/library). Program dengan `package main` wajib memiliki fungsi `main()` sebagai titik awal eksekusi.

2. **import "fmt"** Mengimpor package `fmt` yang berfungsi untuk format input/output.
3. **func main()** Fungsi utama yang akan dijalankan pertama kali ketika program dimulai.
4. **var usia int** untuk menyimpan umur dan **var KK bool** bertipe bool, bernilai true atau false, untuk menunjukkan apakah seseorang memiliki KK.
5. **fmt.Scan(&usia, &KK)** memasukkan dua nilai yaitu usia dan KK(true or false).
6. **if usia >= 17 && KK {fmt.Println("bisa membuat KTP")}**  
**} else {fmt.Println("belum bisa membuat KTP")}** Jika usia  $\geq 17$  dan memiliki KK, maka akan mengeluarkan “bisa membuat KTP”, keduanya harus terpenuhi. Jika salah satunya tidak terpenuhi maka akan mengeluarkan “belum bisa membuat KTP”.

## 2. Guided 2 (HURUF)

### Source code

```
package main

import "fmt"

func main() {
    var kons string
    fmt.Scan(&kons)

    if kons == "A" || kons == "I" ||
        kons == "U" || kons == "E" ||
        kons == "O" || kons == "a" ||
        kons == "i" || kons == "u" ||
        kons == "e" || kons == "o" {
        fmt.Println("Vokal")
    } else if (kons >= "a" && kons <= "z") ||
        (kons >= "A" && kons <= "Z") {
        fmt.Println("Konsonan")
    } else {
        fmt.Println("Bukan huruf")
    }
}
```

## Screenshot program

The screenshot shows a code editor interface with several tabs open. The active tab is 'guided2.go' which contains the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kons string
7     fmt.Scan(&kons)
8
9     if kons == "A" || kons == "I" || kons == "U" || kons == "E" || kons == "O" ||
10    kons == "a" || kons == "i" || kons == "u" || kons == "e" || kons == "o" {
11        fmt.Println("Vokal")
12    } else if (kons >= "a" && kons <= "z") || (kons >= "A" && kons <= "Z") {
13        fmt.Println("Konsonan")
14    } else {
15        fmt.Println("Bukan huruf")
16    }
17 }
```

Below the code editor is a terminal window showing the execution of the program:

```
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided2.go"
A
Vokal
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided2.go"
f
Konsonan
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided2.go"
1
Bukan huruf
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided2.go"
$
```

The terminal also displays the user's name and ID.

## Deskripsi program

Program ini digunakan untuk menentukan apakah input berupa huruf vokal, huruf konsonan, atau bukan huruf.

1. **package main** Menandakan bahwa program ini adalah program utama (bukan package/library). Program dengan package main wajib memiliki fungsi main() sebagai titik awal eksekusi.
2. **import "fmt"** Mengimpor package fmt yang berfungsi untuk format input/output.
3. **func main()** Fungsi utama yang akan dijalankan pertama kali ketika program dimulai.
4. **var kons string** digunakan untuk menampung satu karakter yang diinput pengguna.
5. **fmt.Scan(&kons)** pengguna memasukkan satu karakter
6. **if kons == "A" || kons == "I" || kons == "U" || kons == "E" || kons == "O" || kons == "a" || kons == "i" || kons == "u" || kons == "e" || kons == "o" {**  
    **fmt.Println("Vokal")**  
} **else if (kons >= "a" && kons <= "z") || (kons >= "A" && kons <= "Z") {**  
    **fmt.Println("Konsonan")**  
} **else {**  
    **fmt.Println("Bukan huruf")** } perintah ini digunakan untuk mengecek apakah karakter yang dimasukan adalah vokal, konsonan, atau bukan huruf. Jika input adalah salah satu huruf vokal (A, I, U, E, O baik huruf besar maupun kecil), maka akan mencetak "Vokal", Jika input bukan vokal tetapi masih merupakan huruf alfabet (a-z atau A-Z), maka program mencetak "Konsonan", Jika input bukan huruf alfabet (misalnya angka, simbol, atau karakter lain), program mencetak "Bukan huruf".

## 2 . Guided 3 (URUTAN)

### Source code

```
package main

import "fmt"

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

    a := n / 1000
    b := (n / 100) % 10
    c := (n / 10) % 10
    d := n % 10
    if a < b && b < c && c < d {
        fmt.Println("Terurut membesar")
    } else if a > b && b > c && c > d {
        fmt.Println("Terurut mengecil")
    } else {
        fmt.Println("tidak terurut")
    }
}
```

### Screenshot program

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

- File Explorer:** Shows three files: guided1.go (2 lines), guided2.go (1 line), and guided3.go (1 line).
- Editor:** The guided3.go file is open, displaying the provided Go code.
- Terminal:** The terminal shows the command `PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\guided3.go"` being run, followed by the output:
  - For input 2489: `Terurut membesar`
  - For input 3861: `tidak terurut`
  - For input 9651: `Terurut mengecil`
- Output Window:** A separate window titled "Un..." shows the user information: DIVA ZAHRAH NABILA 109082500112.

### Deskripsi program

Program ini digunakan untuk mengambil 4 angka dan memisahkan angka tersebut lalu mengecek apakah empat digit angka yang diinput pengguna merupakan terurut membesar, terurut mengecil, atau tidak terurut.

1. **package main** Menandakan bahwa program ini adalah program utama (bukan package/library). Program dengan package main wajib memiliki fungsi `main()` sebagai titik awal eksekusi.
2. **import "fmt"** Mengimpor package `fmt` yang berfungsi untuk format input/output.
3. **func main()** Fungsi utama yang akan dijalankan pertama kali ketika program dimulai.
4. **Var n int** digunakan untuk menampung satu karakter yang diinput pengguna.
5. **fmt.Scan(&n)** pengguna memasukan inputan dalam bilangan bulat.
6. **a := n / 1000** (digunakan untuk memisahkan digit pertama pada bilangan ribuan)  
**b := (n / 100) % 10** (digunakan untuk memisahkan digit kedua pada bilangan ratusan)  
**c := (n / 10) % 10** (digunakan untuk memisahkan digit ketiga pada bilangan puluhan)  
**d := n % 10** (digunakan untuk memisahkan digit keempat pada bilangan satuan)
7. **if a < b && b < c && c < d {  
    fmt.Println("Terurut membesar")  
} else if a > b && b > c && c > d {  
    fmt.Println("Terurut mengecil")  
} else {  
    fmt.Println("tidak terurut")}** } digunakan untuk mengecek apakah digit-digit tersebut terurut. Jika terurut dari yang kecil ke besar akan mencetak “Terurut membesar”, Jika digit-digit terurut dari yang terbesar ke kecil maka akan mencetak “Terurut mengecil”. Dan jika digit-digit tersebut tidak berurutan, maka akan mencetak “Tidak terurut”.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var gram int
    fmt.Scan(&gram)

    kg := gram / 1000
    sisa := gram % 1000
    biaya := kg * 10000
    tambahan := 0
    if kg > 10 {
        tambahan = 0
    } else if sisa >= 500 {
        tambahan = sisa * 5
    } else if sisa > 0 {
        tambahan = sisa * 15
    } else {
        tambahan = 0
    }
    total := biaya + tambahan
    fmt.Println("Berat parsel (gram):", gram)
    fmt.Println("Detail berat:", kg, "+", sisa)
    fmt.Println("Detail biaya:", biaya, "+",
    tambahan)
    fmt.Println("Total biaya pengiriman: ", total)
}
```

#### Screenshot program

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

- EXPLORER:** Shows files in the current workspace, including `guided1.go`, `guided2.go`, `guided3.go`, and `soal1.go`.
- CODE EDITOR:** Displays the source code for `soal1.go`. The code calculates shipping costs based on weight (kg) and residue (sisa). It includes logic for different weight ranges and residue amounts.
- TERMINAL:** Shows the command `PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\soal1.go"` and its output:

```
Berat parsel (gram): 6500
Detail berat: 6 + 500
Total biaya: 80000 + 2500
Total biaya pengiriman: 82500
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\soal1.go"
9250
Berat parsel (gram): 9250
Detail berat: 9 + 250
Total biaya: 80000 + 3750
Total biaya pengiriman: 83750
PS D:\LAPRAK\WEEK 10>
```
- AI ASSISTANCE:** A sidebar titled "Build with Agent" provides AI-generated responses and instructions.

## Deskripsi program

Program ini menghitung **biaya pengiriman parsel** berdasarkan berat dalam gram, lalu mengubah menjadi kilogram dan menghitung sisa dari gram itu sendiri, kemudian menghitung biaya tambahan dalam pengiriman.

1. **package main** Menandakan bahwa program ini adalah program utama (bukan package/library). Program dengan package main wajib memiliki fungsi **main()** sebagai titik awal eksekusi.
2. **import "fmt"** Mengimpor package **fmt** yang berfungsi untuk format input/output.
3. **func main()** Fungsi utama yang akan dijalankan pertama kali ketika program dimulai.
4. **var gram int** dan **fmt.Scan(&gram)** Mengambil input berat parsel dalam gram dan menyimpannya pada variabel **gram**.
5. **kg := gram / 1000** untuk mengubah satuan dari gram ke kilogram(Kg).  
**sisa := gram % 1000** untuk menghitung sisa gram.  
**biaya := kg \* 10000** menghitung biaya pengiriman dimana biayanya adalah 10.000 per kilogram.
6. **tambahan := 0** menentukan nilai awal dari **tambahan** adalah 0.  
**if kg > 10 {**  
    **tambahan = 0** (jika total kg lebih dari 10 maka tambahan biayanya 0 atau gratis).  
    **} else if sisa >= 500 {**  
        **tambahan = sisa \* 5** (jika sisa gram >= 500 maka akan dikenakan biaya Rp.5 per gramnya)  
    **} else if sisa > 0 {**  
        **tambahan = sisa \* 15** ((jika sisa gram >0 namun <500 maka akan dikenakan biaya Rp.15 per gramnya)  
    **} else {**  
        **tambahan = 0**} (selain perintah diatas maka biaya tambahannya 0).
7. **total := biaya + tambahan** (menghitung total biaya)
8. **fmt.Println("Berat parsel (gram):", gram)**  
    **fmt.Println("Detail berat:", kg, "+", sisa)**  
    **fmt.Println("Detail biaya:", biaya, "+", tambahan)**  
    **fmt.Println("Total biaya pengiriman: ", total)**  
menampilkan informasi lengkap ke layer.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Scan(&nam)
    if nam >= 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "AB"
    } else if nam > 65 {
        nmk = "B"
    } else if nam > 57.5 {
        nmk = "BC"
    } else if nam > 50 {
        nmk = "C"
    } else if nam > 40 {
        nmk = "D"
    } else if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Nilai Mata Kuliah:", nmk)
}
```

### Screenshot program

The screenshot shows a Go development environment with the following interface elements:

- File Explorer:** Shows files in the current directory: `WEEK 10`, `guided1.go`, `guided2.go`, `guided3.go`, `soal1.go`, and `soal2.go`.
- Code Editor:** Displays the source code for `soal2.go`. The code defines a function `main()` that reads a float value from standard input, determines a grade based on its value, and prints the grade to standard output.
- Terminal:** Shows the command `go run "d:\LAPRAK\WEEK 10\soal2.go"` being run twice, resulting in the output "Nilai Mata Kuliah:A" and "Nilai Mata Kuliah:D".
- Bottom Bar:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS, along with other standard IDE icons.

## Deskripsi program

Program ini berfungsi untuk mengubah nilai akhir mahasiswa berupa angka yang dimasukkan pengguna menjadi nilai huruf berdasarkan rentang penilaian tertentu. Setelah membaca nilai akhir mahasiswa melalui variabel nam, program memeriksa nilainya melalui serangkaian kondisi if dan else if untuk menentukan kategori yang sesuai: nilai 80 ke atas diberi huruf "A", disusul "AB", "B", "BC", "C", dan "D" untuk rentang nilai yang semakin rendah, hingga akhirnya nilai 40 ke bawah akan mendapat huruf "E". Setelah nilai huruf ditentukan dan disimpan dalam variabel nmk, program menampilkan hasil konversi tersebut dengan mencetak "Nilai Mata Kuliah:" diikuti nilai huruf yang diperoleh.

```
1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Println("Nilai akhir mata kuliah: ")
7     fmt.Scan(&nam)
8     if nam > 80 {
9         nam = "A"
10    }
11    if nam > 72.5 {
12        nam = "AB"
13    }
14    if nam > 65 {
15        nam = "B"
16    }
17    if nam > 57.5 {
18        nam = "BC"
19    }
20    if nam > 50 {
21        nam = "C"
22    }
23    if nam > 40 {
24        nam = "D"
25    } else if nam <= 40 {
26        nam = "E"
27    }
28    fmt.Println("Nilai mata kuliah: ", nmk)
29 }
```

Jawablah pertanyaan berikut:

1. Jika **nam** diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?

Jawab: jika code program benar maka saat memasukan angka 80.1 akan menampilkan "A". Namun pada program code di pdf ada beberapa keliruan, sehingga error saat dijalankan.

2. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!

Jawab: Kesahan yang terjadi pada code program yaitu pada baris ke 6 harusnya tidak ada "fmt.Println("Nilai akhir mata kuliah:")" karena akan digunakan pada baris ke 28. Selanjutnya penulisan if kedua dan seterusnya, seharusnya diganti menjadi else if. Jika semuanya if akan terjadi error saat dijalankan. Selanjutnya pada keluaran setelah kondisi if(baris 9,12,15,18,21,24,26) seharusnya variabel **nmk** bukan **nam**.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var bil, faktor int
    fmt.Scan(&bil)
    faktor = 0

    for i := 1; i <= bil; i++ {
        if bil%i == 0 {
            fmt.Printf("%d ", i)
            faktor++
        }
    }
    fmt.Println()

    if faktor == 2 {
        fmt.Println("Prima:true")
    } else if faktor > 2 {
        fmt.Println("Prima:false")
    } else {
        fmt.Println("Prima:false")
    }
}
```

#### Screenshot program

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

- Project Structure:** WEEK 10 contains files: guided1.go, guided2.go, guided3.go, soal1.go, soal2.go, and soal3.go.
- Code Editor:** The code for soal3.go is displayed, showing a function main() that reads an integer from standard input, counts its divisors, and prints the result as either Prima:true, Prima:false, or Prima:false based on the divisor count.
- Terminal:** The terminal window shows the command "go run" being run on the file "soal3.go". The output is:

```
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\soal3.go"
12
1 2 3 4 6 12
Prima:false
PS D:\LAPRAK\WEEK 10> go run "d:\LAPRAK\WEEK 10\soal3.go"
7
1 7
Prima:true
PS D:\LAPRAK\WEEK 10>
```
- Code View:** A floating window shows the code for soal3.go, with the cursor on the line "faktor++". The code is identical to the one in the code editor.
- Status Bar:** The status bar at the bottom right indicates "Ln 5, Col 14 Tab Size: 4 UTF-8".

## Deskripsi program

Program ini digunakan untuk menampilkan faktor-faktor dari suatu bilangan lalu menentukan apakah bilangan tersebut bilangan prima atau bukan.

1. **package main** Menandakan bahwa program ini adalah program utama (bukan package/library). Program dengan package main wajib memiliki fungsi main() sebagai titik awal eksekusi.
2. **import "fmt"** Mengimpor package fmt yang berfungsi untuk format input/output.
3. **func main()** Fungsi utama yang akan dijalankan pertama kali ketika program dimulai.
4. **var bil, faktor int** dan **fmt.Scan(&bil)** Mengambil input berupa bilangan bulat dan menyimpannya pada variabel bil.
5. **faktor = 0** Menentukan nilai awal faktor yaitu 0.  
**for i := 1; i <= bil; i++ {**  
    **if bil%i == 0 {**  
        **fmt.Printf("%d ", i)**  
        **faktor++ }**  kode perulangan ini digunakan untuk mencari faktor dari suatu bilangan.
5. **if faktor == 2 {**  
    **fmt.Println("Prima:true")** jika faktornya berjumlah dua,maka bilangan tersebut prima dan akan mencetak "Prima : true".  
} **else if faktor > 2 {**  
    **fmt.Println("Prima:false")** jika faktornya lebih dari dua,maka bilangan tersebut bukan prima dan akan mencetak "Prima : false".  
} **else {**  
    **fmt.Println("Prima:false") }** selain dari perintah diatas akan mencetak "Prima : false".