

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 10

ELSE-IF



Disusun oleh:

SATRIYA WAHYU PRAKOSO

109082500219

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 usia int

    var kk bool

    fmt.Scan(&usia, &kk)

    if usia >= 17 && kk {

        fmt.Println("bisa membuat KTP")

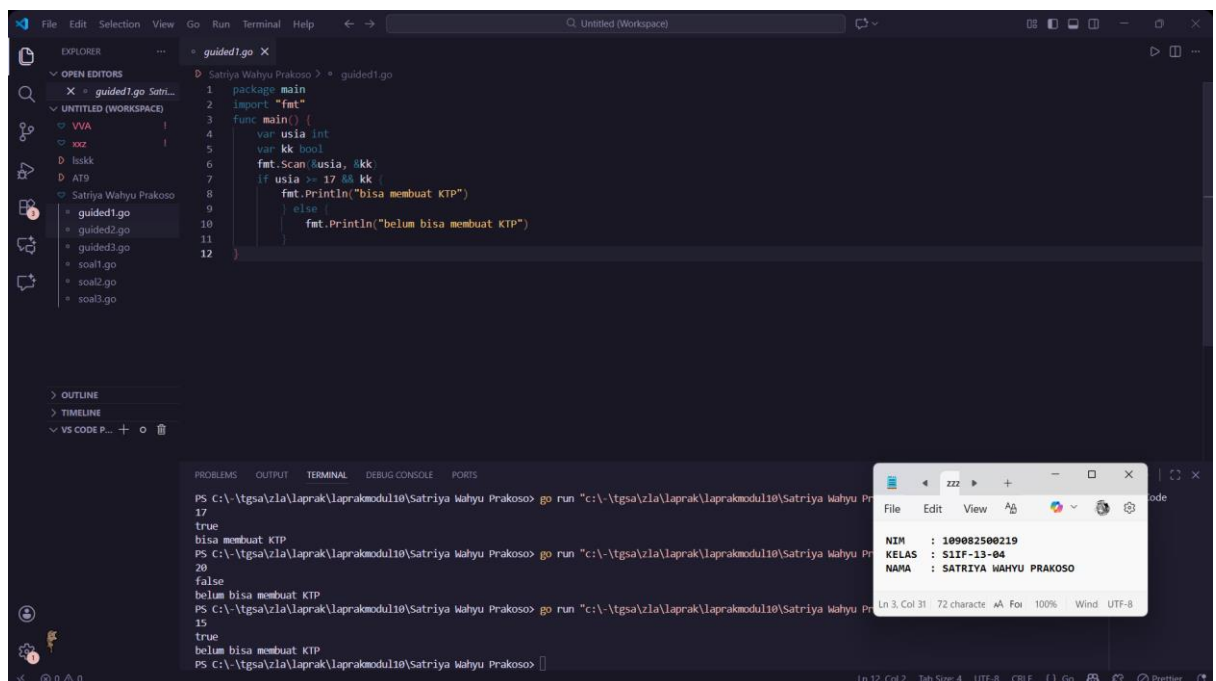
    } else {

        fmt.Println("belum bisa membuat KTP")

    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan apakah seorang penduduk bisa membuat KTP atau tidak. Program ditulis dengan bahasa Go.

Var `usia` `int` digunakan untuk menyimpan input usia yang akan dimasukkan oleh pengguna dengan tipe `int`, sedangkan var `kk` `bool` digunakan untuk menyimpan input apakah memiliki `kk` atau tidak dalam tipe `boolean`. Setelah input dimasukkan input akan discan, jika input usia yang dimasukkan lebih dari sama dengan 17 dan `kk` adalah `true` maka program akan mencetak output berupa “bisa membuat KTP”, selain itu jika salah 1 dari kondisi tersebut salah misalnya usia dibawah 17 dan `kk` ada, atau usia dibawah 17 dan `kk` tidak ada maka program akan mencetak output berupa “belum bisa membuat KTP”.

2. Guided 2

Source Code

```
package main

import "fmt"

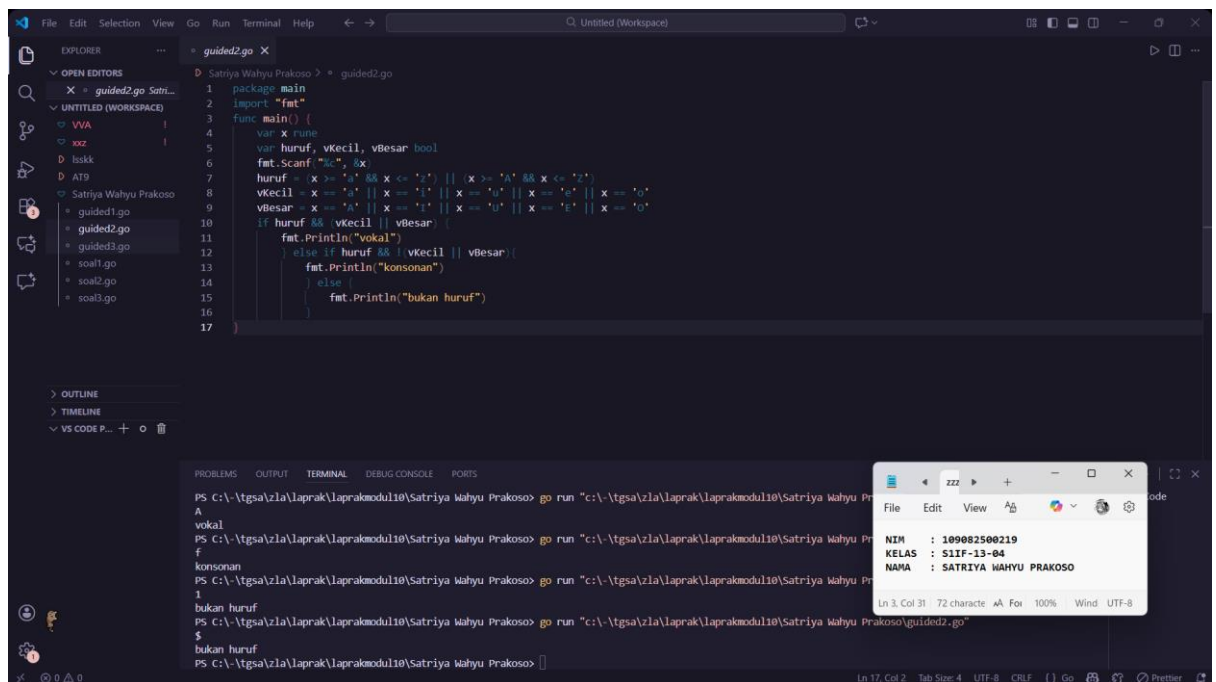
func main() {
    var x rune
    var huruf, vKecil, vBesar bool
    fmt.Scanf("%c", &x)

    huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')

    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'

    if huruf && (vKecil || vBesar) {
        fmt.Println("vokal")
    } else if huruf && !(vKecil || vBesar) {
        fmt.Println("konsonan")
    } else {
        fmt.Println("bukan huruf")
    }
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan suatu alfabet yang diberikan adalah vokal atau konsonan. Program ditulis dengan bahasa GO.

Var x rune digunakan untuk menyimpan input yang dimasukkan pengguna dalam tipe rune. Var huruf, vKecil, vBesar bool digunakan untuk menyimpan hasil rumus-rumus nanti dalam tipe boolean. %c digunakan untuk mengambil 1 karakter yang di input, setelah input dimasukkan input di scan lalu masuk dalam bagian penentuan yaitu: untuk menentukan variabel huruf adalah true atau false menggunakan **huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')**, untuk menentukan huruf vokal kecil atau bukan menggunakan **vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'**, dan untuk menentukan apakah huruf vokal besar atau bukan menggunakan **vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'**. Setelah ini kita menentukan outputnya dengan cara jika variabel huruf dan vKecil/vBesar adalah sama-sama true maka program akan menampilkan output berupa “vokal”, jika variabel huruf true dan variabel vKecil/vBesar false maka output akan menampilkan “konsonan”, lalu selain itu misalnya variabel huruf dan vKecil/vBesar sama-sama false maka output akan menampilkan “bukan huruf”.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var bilangan, d1, d2, d3, d4 int
```

```

var teks string

fmt.Print("Bilangan: ")

fmt.Scan(&bilangan)

d4 = bilangan % 10

d3 = (bilangan % 100) / 10

d2 = (bilangan % 1000) / 10

d1 = bilangan / 1000

if d1 < d2 && d2 < d3 && d3 < d4 {

    teks = "terurut membesar"

}else if d1 > d2 && d2 > d3 && d3 > d4{

    teks = "terurut mengecil"

    } else {

        teks = "tidak terurut"

    }

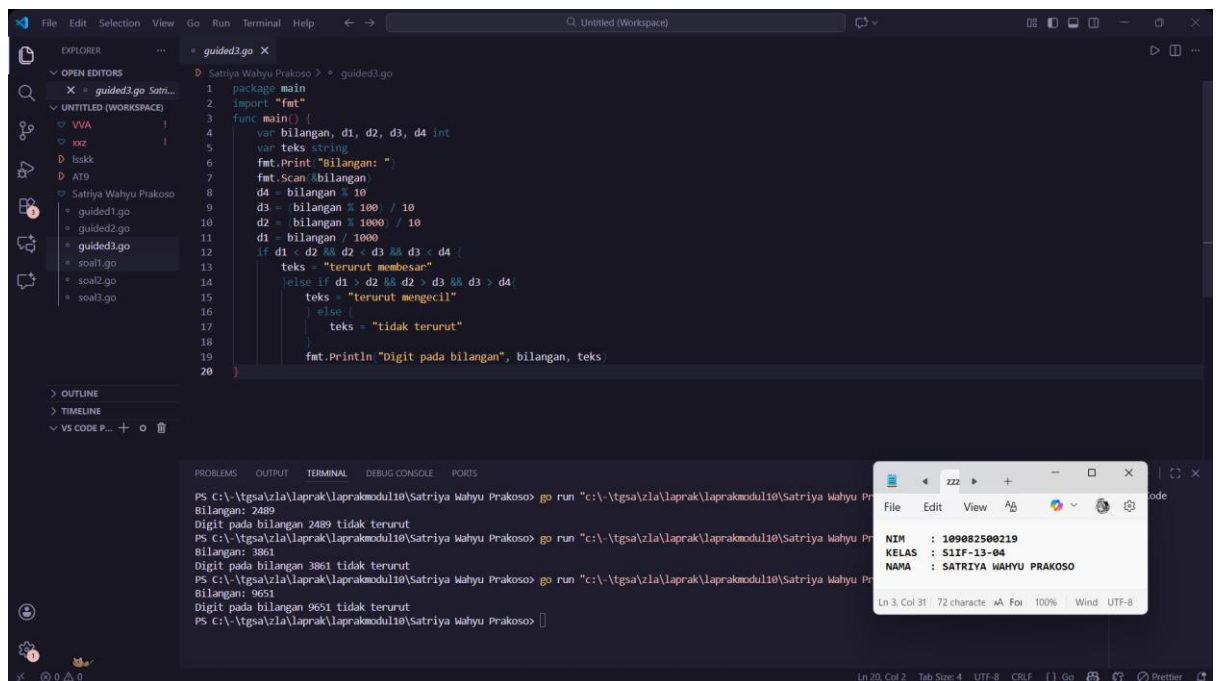
    fmt.Println("Digit pada bilangan",

bilangan, teks)

}

```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan apakah digit dalam suatu bilangan terurut membesar, mengecil, atau tidak terurut. Program ditulis dengan bahasa GO.

Var bilangan, d1, d2, d3, d4 int digunakan untuk menyimpan data-data dalam tipe int, bilangan untuk input dan d1,d2,d3,d4 untuk menyimpan hasil rumus-rumus perhitungan nanti. Var teks string digunakan untuk menyimpan hasil dari if nanti dalam tipe string. **fmt.Print("Bilangan: ")** digunakan agar pengguna tau kalau harus memasukkan bilangan. Setelah input dimasukkan input akan di scan lalu program akan menentukan nilai d4, d3, d2,d1 dengan cara **d4 = bilangan % 10, d3 = (bilangan % 100) / 10, d2 = (bilangan % 1000) / 10, d1 = bilangan / 1000**. Jika d1 lebih kecil dari d2, d2 lebih kecil dari d3, d3 lebih kecil dari d4 maka variabel teks akan berubah menjadi "terurut membesar", jika d1 lebih besar dari d2, d2 lebih besar dari d3, d3 lebih besar dari d4 maka variabel teks akan berubah menjadi "terurut mengecil", jika kedua if sebelumnya tidak terpenuhi maka teks akan berubah menjadi "tidak terurut". Setelah ini output akan menampilkan hasilnya berupa "Digit pada bilangan", bilangan yang dimasukkan dan variabel teks setelah semua tadi.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var beratawal, kilo, sisa int64
    var totalkilo, totalsisa, totalakhir int64
    fmt.Scan(&beratawal)
    kilo = beratawal / 1000
    totalkilo = kilo * 10000
    sisa = beratawal - (kilo * 1000)
    if sisa >= 500 {
        totalsisa = sisa * 5
    } else {
        totalsisa = sisa * 15
    }
}
```

```

        if kilo > 10 {
            totalakhir = totalkilo
        } else {
            totalakhir = totalkilo + totalsisa
        }

        fmt.Println("Berat parcel (gram) : ", beratawal)

        fmt.Printf("Detail berat : %d kg + %d gr \n", kilo,
            sisa)

        fmt.Printf("Detail biaya : Rp. %d + Rp. %d \n",
            totalkilo, totalsisa)

        fmt.Print("Total biaya : Rp. ", totalakhir)
    }

```

Screenshoot program

The screenshot shows a Go program in VS Code. The source code in `soal1.go` defines a `main` function that calculates the total weight and cost of a parcel based on its weight in kilograms and grams. The program uses `fmt` for input/output and `int64` for calculations.

```

1 package main
2 import "fmt"
3 func main() {
4     var beratawal, kilo, sisa int64
5     var totalkilo, totalsisa, totalakhir int64
6     fmt.Scan(&beratawal)
7     kilo = beratawal / 1000
8     totalkilo = kilo * 1000
9     sisa = beratawal - kilo * 1000
10    if sisa >= 500 {
11        totalsisa = sisa * 5
12    } else {
13        totalsisa = sisa * 15
14    }
15    if kilo > 10 {
16        totalakhir = totalkilo
17    } else {
18        totalakhir = totalkilo + totalsisa
19    }
20    fmt.Println("Berat parcel (gram) : ", beratawal)
21    fmt.Printf("Detail berat : %d kg + %d gr \n", kilo, sisa)
22    fmt.Printf("Detail biaya : Rp. %d + Rp. %d \n", totalkilo, totalsisa)
23    fmt.Print("Total biaya : Rp. ", totalakhir)
24 }

```

The terminal output shows the program being run twice. The first run uses input `8500`, resulting in a total weight of 8 kg + 500 gr and a total cost of Rp. 82500. The second run uses input `9250`, resulting in a total weight of 9 kg + 250 gr and a total cost of Rp. 93750.

```

PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso> go run "c:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso\soal1.go"
8500
Berat parcel (gram) : 8500
Detail berat : 8 kg + 500 gr
Detail biaya : Rp. 80000 + Rp. 2500
Total biaya : Rp. 82500
PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso> go run "c:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso\soal1.go"
9250
Berat parcel (gram) : 9250
Detail berat : 9 kg + 250 gr
Detail biaya : Rp. 90000 + Rp. 3750
Total biaya : Rp. 93750
PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso>

```

A small window in the bottom right corner displays the following information:

```

NIM : 109082500219
KELAS : SIIF-13-04
NAMA : SATRIYA WAHYU PRAKOSO

```

```
1 package main
2 import "fmt"
3 func main() {
4     var beratawal, kilo, sisa int64
5     var totalkilo, totalsisa, totalakhir int64
6     fmt.Scan(&beratawal)
7     kilo = beratawal / 1000
8     totalkilo = kilo * 10000
9     sisa = beratawal - kilo * 1000
10    if sisa >= 500 {
11        totalsisa = sisa * 5
12    } else {
13        totalsisa = sisa * 15
14    }
15    if kilo > 10 {
16        totalakhir = totalkilo
17    } else {
18        totalakhir = totalkilo + totalsisa
19    }
20    fmt.Println("Berat parcel (gram) : ", beratawal)
21    fmt.Printf("Detail berat : %d kg + %d gr \n", kilo, sisa)
22    fmt.Printf("Detail biaya : Rp. %d + Rp. %d \n", totalkilo, totalsisa)
23    fmt.Print("Total biaya : Rp. ", totalakhir)
24 }
```

PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso> go run "c:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso\soal1.go"

9250
Berat parcel (gram) : 9250
Detail berat : 9 kg + 250 gr
Detail biaya : Rp. 90000 + Rp. 3750
Total biaya : Rp. 93750

PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso> go run "c:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso\soal2.go"

11750
Berat parcel (gram) : 11750
Detail berat : 11 kg + 750 gr
Detail biaya : Rp. 110000 + Rp. 3750
Total biaya : Rp. 110000

PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso>

Deskripsi program

Program ini digunakan untuk menghitung biaya kirim berdasarkan berat parcel. Program ditulis dengan bahasa GO.

var beratawal, kilo, sisa int64 dan var totalkilo, totalsisa, totalakhir int64 digunakan untuk menyimpan data-data dalam tipe int. variabel beratawal digunakan untuk menyimpan input. Setelah input dimasukkan input akan discan. nilai dari variabel kilo adalah input dibagi 1000, nilai variabel totalkilo adalah variabel kilo dikali 10000, nilai variabel sisa adalah input dikurangi (kilo dikali 1000). Jika variabel sisa lebih besar sama dengan 500 maka varibel totalsisa adalah sisa dikali 5, selain itu(dibawah 500) maka variabel totalsisa adalah sisa dikali 15. Jika varibel kilo lebih dari 10 maka nilai variabel totalakhir sama dengan variabel totalkilo, selain itu(dibawah 10) maka nilai variabel totalakhir adalah variabel totalkilo ditambah totalsisa. Setelah ini output akan menampilkan hasilnya, di baris pertama menampilkan berat awal, dibaris kedua menampilkan detail berat berapa kg(variabel kilo) ditambah berapa gr(variabel sisa), dibaris ketiga menampilkan detail biaya variabel kilo(totalkilo) ditambah biaya variabel sisa(totalsisa), di baris keempat menampillan total biaya penjumlahan(totalakhir).

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var nam float64
```



```
var nmk string
fmt.Print("Nilai akhir mata kuliah: ")
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 {
    nmk = "E"
}
fmt.Println("Nilai mata kuliah:", nmk)
}
```

Screenshoot program

The screenshot shows a VS Code editor with a Go file named `soal2.go`. The code defines a `main` package with a `main` function. It declares a `nam` variable of type `Float64` and an `nmk` variable of type `string`. The program prompts the user for "Nilai akhir mata kuliah:" and scans the input. It uses a series of `if` and `else if` statements to assign a grade (`nmk`) based on the input value: `nmk = "A"` for values ≥ 80 , `nmk = "AB"` for $72.5 \leq \text{value} < 80$, `nmk = "B"` for $65 \leq \text{value} < 72.5$, `nmk = "BC"` for $57.5 \leq \text{value} < 65$, `nmk = "C"` for $50 \leq \text{value} < 57.5$, `nmk = "D"` for $40 \leq \text{value} < 50$, and `nmk = "E"` for values < 40 . Finally, it prints the result using `fmt.Println("Nilai mata kuliah:", nmk)`.

The terminal at the bottom shows the execution of the program with the following output:

```
PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso> go run "c:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso\soal2.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso> go run "c:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso\soal2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso> go run "c:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso\soal2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\-tgsa\zla\laprak\laprakmodul10\Satriya Wahyu Prakoso>
```

A small window in the foreground displays the following information:

```
NIM : 109082500219
KELAS : SLIF-13-04
NAMA : SATRIYA WAHYU PRAKOSO
```

Deskripsi program

Program ini digunakan untuk menentukan NMK dari NAM yang dimasukkan. Program ditulis dengan bahasa GO.

Var `nam float64` digunakan untuk menyimpan data input yang dimasukkan dalam tipe `float`. Var `nmk string` digunakan untuk menyimpan hasil `if` nanti dalam tipe `string`. `fmt.Print("Nilai akhir mata kuliah: ")` digunakan agar pengguna tau kalau harus memasukkan NAM, setelah input dimasukkan input akan discan. Jika input lebih dari 80 maka variabel `nmk` berubah menjadi A, kalau input lebih besar dari 72,5 menjadi AB, kalau input lebih besar dari 65 menjadi B, kalau input lebih besar dari 57,5 menjadi BC, kalau input lebih besar dari 50 menjadi C, kalau input lebih besar dari 40 menjadi D, dan selain semua itu menjadi E. Setelah itu output akan menampilkan hasil variabel `nmk`.

ini adalah program setelah diperbaiki, hasil sebelum program diperbaiki mempunyai beberapa kesalahan.

```

1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Print("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: keluaran setelah kode diperbaiki menampilkan "A". Sebelum kode diperbaiki output yang ditampilkan keliru.

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

Jawab: **Kesalahan pertama** yaitu pada baris keenam, sebaiknya semua yang ada di baris ini tidak ada dan dihapus karena ini harusnya digunakan pada baris 28.

Kesalahan kedua yaitu tidak menggunakan else pada bagian if ke 2 dan seterusnya, jika tidak menggunakan else outputnya akan salah dan tidak sesuai yang diinginkan.

Kesalahan ketiga adalah kekeliruan dalam baris 9,12,15,18,21,24,26 di baris-baris ini harusnya nmk bukanlah nam.

3. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5.

Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Jawab: source code dan screenshot program yang saya masukkan tadi adalah program yang telah diperbaiki

3. Tugas 3

Source code

```

package main

import "fmt"

func main() {

    var bil, i, faktor int

```

```

var prima bool = true

fmt.Scan(&bil)

fmt.Println("Bilangan : ", bil)

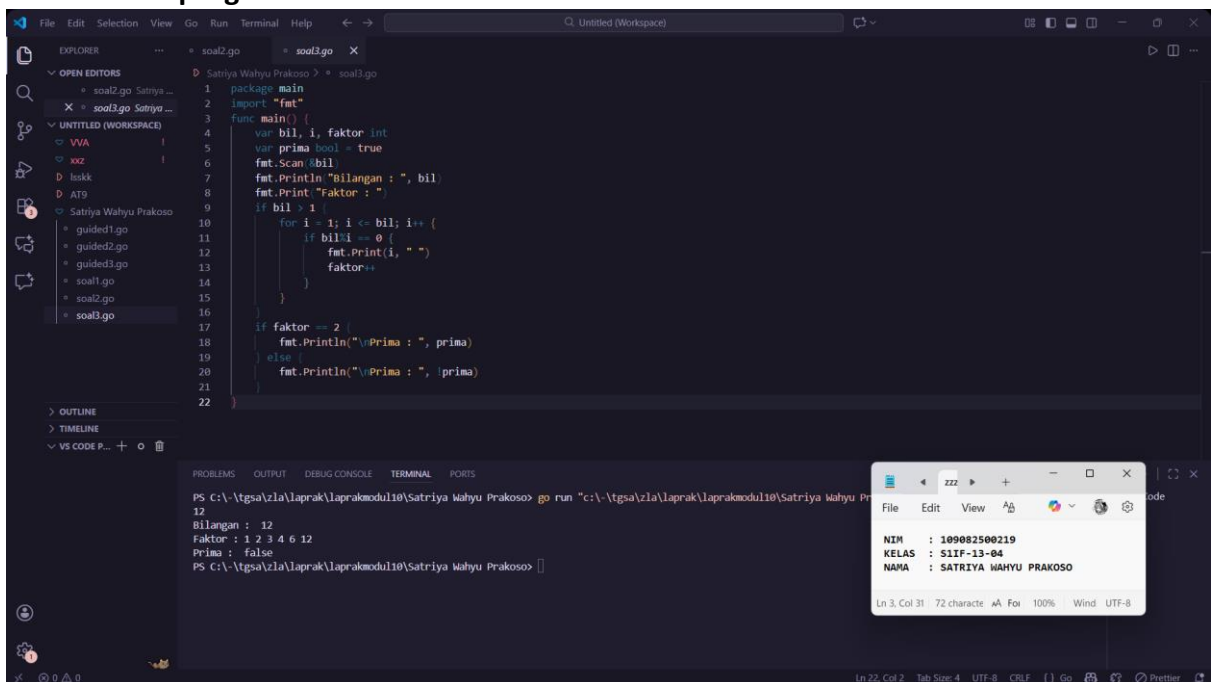
fmt.Print("Faktor : ")

if bil > 1 {
    for i = 1; i <= bil; i++ {
        if bil%i == 0 {
            fmt.Print(i, " ")
            faktor++
        }
    }
}

if faktor == 2 {
    fmt.Println("\nPrima : ", prima)
} else {
    fmt.Println("\nPrima : ", !prima)
}
}

```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menghitung faktor-faktor dari sebuah bilangan dan menentukan apakah bilangan tersebut merupakan bilangan prima atau bukan.

Program ditulis dengan bahasa GO.

Var bil, im faktor int digunakan untuk menyimpan data-data dalam tipe int, variabel bil digunakan untuk menyimpan input yang dimasukkan. Var prima bool digunakan untuk output nanti, disini kita langsung set menjadi true.

Setelah input dimasukkan program akan mencetak bilangan yang dimasukkan dengan **fmt.Println("Bilangan : ", bil)**, setelah ini input akan masuk ke bagian perulangan, jika input lebih dari 1 i akan di set menjadi 1, selama i kurang dari sama dengan bil(input), i ditambah 1 setiap perulangan, di setiap perulangan jika bil(input) di modulus i sama dengan 0 maka faktor dari bil akan akan dicetak dengan **fmt.Print(i, " ")**. Setiap faktor ditemukan, variabel faktor ditambah 1.

Setelah perulangan selesai jika faktor sama dengan 2 maka output akan menampilkan Prima yaitu true karena pada awal tadi variabel prima di set true. Selain itu(jika faktor bukan sama dengan 2 maka output akan menampilkan hasil Prima yaitu false karena menggunakan !prima, ! berarti not jadi true berubah menjadi false.