

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE-IF



Disusun oleh:

RAFIF ARFA DHIAPRAJA

109082500041

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

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.Println("Masukkan usia Anda: ")
    fmt.Scan(&usia)
    fmt.Println("Apakah Anda memiliki Kartu Keluarga (true/false): ")
    fmt.Scan(&KK)

    if usia >= 17 && KK == true {
        fmt.Println("Bisa Membuat KTP")
    } else {
        fmt.Println("Tidak Bisa Membuat KTP")
    }
}
```

Screenshoot program

The screenshot shows a Visual Studio Code interface with the following components:

- Editor:** Displays the Go code for "ifelse.go". The code defines a main function that prompts the user for age and family card status, then prints whether KTP can be made based on the input.
- Terminal:** Shows the command "go run ifelse.go" being run three times from the directory "C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10". The terminal output includes user inputs and the resulting messages ("Bisa Membuat KTP" or "Tidak Bisa Membuat KTP").
- Output:** A modal window titled "NAMA" showing the user's details: NAMA : RAFIF ARFA DHIAPRAJA, KELAS : S1IF-13-07, and NIM : 109082500041.
- Bottom Status Bar:** Shows the current file path, line count (Ln 3, Col 19), character count (65 caracte), and other settings like font and encoding.

Deskripsi program

Pada program ini, saya membuat sebuah program untuk menentukan apakah seseorang bisa membuat KTP atau tidak. Program ini menggunakan dua input dari pengguna, yaitu usia dan kepemilikan Kartu Keluarga (KK). Selain itu, digunakan satu kondisi logika untuk mengevaluasi apakah syarat pembuatan KTP telah terpenuhi.

Pertama, program meminta pengguna memasukkan usia dengan menggunakan `fmt.Scan(&usia)`. Nilai usia ini digunakan sebagai syarat utama karena seseorang hanya dapat membuat KTP jika sudah berusia 17 tahun atau lebih. Setelah itu, program meminta input kedua berupa nilai boolean true atau false yang menyatakan apakah pengguna memiliki Kartu Keluarga. Input ini disimpan dalam variabel KK. Setelah kedua data diterima, program melakukan pengecekan menggunakan kondisi `if usia >= 17 && KK == true`. Kondisi ini bermakna bahwa pengguna dapat membuat KTP hanya jika dua syarat terpenuhi secara bersamaan, yaitu usia minimal 17 tahun dan memiliki Kartu Keluarga. Jika kedua syarat benar, program menampilkan pesan "Bisa Membuat KTP".

Namun, jika salah satu atau kedua syarat tidak terpenuhi, program masuk ke bagian else dan menampilkan pesan "Tidak Bisa Membuat KTP".

Dengan demikian, program ini secara sederhana mengevaluasi persyaratan administratif pembuatan KTP berdasarkan usia dan kepemilikan KK, serta memberikan output sesuai kondisi yang terpenuhi oleh pengguna.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var huruf string
    fmt.Print("Masukkan sebuah huruf: ")
    fmt.Scan(&huruf)

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

Screenshot program

The screenshot shows a Go development environment with three tabs open: 'ifelse2.go 2', 'ifelse2.go 1', and 'ifelse3.go 1'. The code in 'ifelse2.go 2' is as follows:

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

The 'TERMINAL' tab shows the execution of the program:

```
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10> go run ifelse2.go
Masukkan sebuah huruf: A
vokal
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10> go run ifelse2.go
Masukkan sebuah huruf: f
konsonan
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10> go run ifelse2.go
Masukkan sebuah huruf: 1
bukan huruf
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10> go run ifelse2.go
Masukkan sebuah huruf: $
bukan huruf
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10>
```

The 'OUTPUT' tab shows the application window with the input 'NAMA : RAFIF ARFA DHIAPRAJA', 'KELAS : S1IF-13-07', and 'NIM : 109082500041'.

Deskripsi program

Pada program ini, saya membuat sebuah program untuk menentukan apakah sebuah karakter yang dimasukkan pengguna merupakan huruf vokal, huruf konsonan, atau bukan huruf. Program menggunakan struktur percabangan if – else if – else untuk memeriksa setiap kondisi yang mungkin terjadi.

Pertama, program meminta pengguna memasukkan sebuah karakter melalui `fmt.Scan(&huruf)`. Nilai ini disimpan dalam variabel `huruf` bertipe `string`. Setelah input diterima, program langsung melakukan pemeriksaan berdasarkan tiga kategori.

Pada kondisi pertama, program memeriksa apakah karakter tersebut merupakan huruf vokal. Pemeriksaan dilakukan dengan mencocokkan input terhadap semua huruf vokal, baik huruf besar (A, I, U, E, O) maupun huruf kecil (a, i, u, e, o). Jika salah satu kondisi terpenuhi, maka program akan menampilkan output "vokal".

Jika bukan vokal, program lanjut memeriksa apakah input tersebut masih merupakan huruf alfabet. Pengecekan dilakukan dengan membandingkan rentang karakter menggunakan kondisi `(huruf >= "a" && huruf <= "z")` atau `(huruf >= "A" && huruf <= "Z")`. Jika kondisi ini terpenuhi, maka karakter tersebut adalah huruf alfabet tetapi bukan vokal, sehingga program menampilkan "konsonan".

Apabila kedua kondisi sebelumnya tidak terpenuhi, program menganggap bahwa karakter yang dimasukkan bukan merupakan huruf. Dalam hal ini, program akan menampilkan output "bukan huruf".

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3, d4 int
    fmt.Print("Masukkan sebuah bilangan 4 digit: ")
    fmt.Scan(&bilangan)

    d1 = bilangan / 1000
    d2 = (bilangan / 100) % 10
    d3 = (bilangan / 10) % 10
    d4 = bilangan % 10

    if d1 < d2 && d2 < d3 && d3 < d4 {
        fmt.Println("Bilangan tersebut naik")
    } else if d1 > d2 && d2 > d3 && d3 > d4 {
        fmt.Println("Bilangan tersebut turun")
    } else {
    }
}
```

```

        fmt.Println("Bilangan tersebut tidak naik dan
        tidak turun")

    }

}

```

Screenshoot program

```

1 Welcome   ifelse.go 2   ifelse2.go 1   ifelse3.go 1
2 ifelse3.go > main
3
4
5 func main() {
6     var bilangan, d1, d2, d3, d4 int
7     fmt.Print("Masukkan sebuah bilangan 4 digit: ")
8     fmt.Scan(&bilangan)
9
10    d1 = bilangan / 1000
11    d2 = (bilangan / 100) % 10
12    d3 = (bilangan / 10) % 10
13    d4= bilangan % 10
14
15    if d1 < d2 && d2 < d3 && d3 < d4 {
16        fmt.Println("Bilangan tersebut naik")
17    } else if d1 > d2 && d2 > d3 && d3 > d4 {
18        fmt.Println("Bilangan tersebut turun")
19    } else {
20        fmt.Println("Bilangan tersebut tidak naik dan tidak turun")
21    }
22 }

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10> go run ifelse3.go
Masukkan sebuah bilangan 4 digit: 2489
Bilangan tersebut naik
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10> go run ifelse3.go
Masukkan sebuah bilangan 4 digit: 3861
Bilangan tersebut turun
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10> go run ifelse3.go
Masukkan sebuah bilangan 4 digit: 9651
Bilangan tersebut turun
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\IF Minggu 10>

File Edit View H1 ... NAMA arfa-dr ...
NAMA : RAFIF ARFA DHIAPRAJA
KELAS : S1IF-13-07
NIM : 109082500041

```

Deskripsi program

Pada program ini, saya membuat program untuk menentukan apakah sebuah bilangan 4 digit termasuk kategori **naik**, **turun**, atau **tidak keduanya**. Bilangan dikatakan *naik* jika setiap digitnya semakin membesar dari kiri ke kanan, dan dikatakan *turun* jika setiap digitnya semakin mengecil dari kiri ke kanan.

Pertama, saya mendeklarasikan empat variabel, yaitu d1, d2, d3, dan d4, yang masing-masing digunakan untuk menyimpan digit pertama hingga digit keempat dari bilangan yang dimasukkan. Setelah itu, program menampilkan pesan "Masukkan sebuah bilangan 4 digit:" agar pengguna memasukkan angka yang sesuai. Nilai bilangan kemudian dibaca menggunakan `fmt.Scan(&bilangan)`.

Selanjutnya, saya memecah bilangan tersebut menjadi empat digit.

- d1 dihitung dengan `bilangan / 1000` untuk mengambil digit ribuan.
- d2 dihitung dengan `(bilangan / 100) % 10` untuk mengambil digit ratusan.

- d3 dihitung dengan $(\text{bilangan} / 10) \% 10$ untuk mengambil digit puluhan.
- d4 dihitung dengan $\text{bilangan} \% 10$ untuk mengambil digit satuan.

Setelah semua digit diperoleh, program melakukan pengecekan menggunakan struktur if – else if – else.

- Pada kondisi pertama, jika $d1 < d2$, $d2 < d3$, dan $d3 < d4$, maka digit-digit tersebut semakin membesar, sehingga program menampilkan “**Bilangan tersebut naik**”.
- Pada kondisi kedua, jika $d1 > d2$, $d2 > d3$, dan $d3 > d4$, maka digit-digit semakin mengecil, sehingga program menampilkan “**Bilangan tersebut turun**”.
- Jika kedua kondisi tersebut tidak terpenuhi, berarti digit-digitnya tidak berurutan naik ataupun turun, sehingga program menampilkan “**Bilangan tersebut tidak naik dan tidak turun**”.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var berat,hargaSisa int
    fmt.Scan(&berat)

    kg := berat / 1000
    sisa := berat % 1000

    hargaKg := kg * 10000

    if kg > 10 {
        hargaSisa = 0
    } else if sisa >= 500 {
```

```

        hargaSisa = sisa * 5

    } else if sisa < 500 {

        hargaSisa = sisa * 15

    }

totalHarga := hargaKg + hargaSisa

fmt.Println("Detail Berat:", kg, "kg", "+", sisa, "gr")

fmt.Println("Detail Biaya: Rp.", hargaKg, "+ Rp.",
hargaSisa)

fmt.Println("Total Biaya: Rp.", totalHarga)

}

```

Screenshot program

The screenshot shows a code editor with the file `Unguided1.go` open. The code implements a function `main()` that calculates the total cost based on weight (`kg`) and remaining weight (`sisa`). It uses conditional statements to determine the price per kilogram for different weight ranges. The terminal window to the right shows the execution of the program and its output, which includes the breakdown of costs (kg at 10000Rp/kg and sisa at 2500Rp/kg) and the total cost.

```

Welcome ➜ Unguided1.go 2 ➜ Unguided2.go 1 ➜ Unguided3.go 1
Unguided1.go > main
4
5 func main() {
6     var berat,hargaSisa int
7     fmt.Scan(&berat)
8
9     kg := berat / 1000
10    sisa := berat % 1000
11
12    hargaKg := kg * 10000
13
14    if kg > 10 {
15        hargaSisa = 0
16    } else if sisa >= 500 {
17        hargaSisa = sisa * 5
18    } else if sisa < 500 {
19        hargaSisa = sisa * 15
20    }
21
22    totalHarga := hargaKg + hargaSisa
23
24    fmt.Println("Detail Berat:", kg, "kg", "+", sisa, "gr")
25    fmt.Println("Detail Biaya: Rp.", hargaKg, "+ Rp.", hargaSisa)
26    fmt.Println("Total Biaya: Rp.", totalHarga)
27 }

```

NAMA : RAFIF ARFA DHIPRAJA
KELAS : S1IF-13-07
NIM : 109082500041

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided1.go
8500
Detail Berat: 8 kg + 500 gr
Detail Biaya: Rp. 80000 + Rp. 2500
Total Biaya: Rp. 82500
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided1.go
9250
Detail Berat: 9 kg + 250 gr
Detail Biaya: Rp. 90000 + Rp. 3750
Total Biaya: Rp. 93750

Deskripsi program

Pada program ini, saya membuat sebuah program untuk menghitung total biaya berdasarkan berat suatu barang yang dimasukkan dalam satuan gram. Pertama, pengguna memasukkan berat barang, kemudian program memisahkan berat tersebut menjadi dua bagian, yaitu kilogram dan sisa gramnya. Nilai kilogram diperoleh dari hasil pembagian berat / 1000, sedangkan sisa gram diperoleh dari berat % 1000.

Setelah berat dipisahkan, program menghitung biaya untuk bagian kilogram dengan mengalikan jumlah kilogram dengan harga per kilogram, yaitu Rp 10.000. Hasil perhitungan ini disimpan dalam variabel hargaKg.

Selanjutnya, program menentukan biaya tambahan untuk sisa gram dengan beberapa kondisi. Jika jumlah kilogram lebih dari sepuluh, maka gram sisa digratiskan biayanya, sehingga hargaSisa diisi dengan nilai 0. Jika sisa gram lebih dari atau sama dengan 500 gram, biaya dihitung dengan tarif Rp 5 per gram. Namun, jika sisa gram kurang dari 500, biaya dihitung dengan tarif Rp 15 per gram. Nilai biaya tambahan ini disimpan dalam variabel hargaSisa.

Total biaya didapat dengan menjumlahkan hargaKg dan hargaSisa, kemudian program menampilkan detail berat dalam bentuk kilogram dan gram, detail biaya dari masing-masing bagian, serta total biaya keseluruhannya.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string

    fmt.Println("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 if nam <= 40 {

nmk = "E"

}

fmt.Println("Nilai mata kuliah: ",nmk)

}
```

Screenshoot program

```
Unguided2.go > main
5 func main() {
7     var nmk string
8
9     fmt.Println("Nilai akhir mata kuliah: ")
10    fmt.Scan(&nam)
11
12    if nam > 80 {
13        nmk = "A"
14    } else if nam > 72.5 {
15        nmk = "AB"
16    } else if nam > 65 {
17        nmk = "B"
18    } else if nam > 57.5 {
19        nmk = "BC"
20    } else if nam > 50 {
21        nmk = "C"
22    } else if nam > 40 {
23        nmk = "D"
24    } else if nam <= 40 {
25        nmk = "E"
26    }
27
28    fmt.Println("Nilai mata kuliah: ", nmk)
```

NAMA : RAFIF ARFA DHIAPRAJA
KELAS : S1IF-13-07
NIM : 109082500041

File Edit View H1 ... 🌐 ⚙️

Ln 3, Col 19 65 character ⌂ Form 100% Windows UTF-8

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided2.go
Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: A
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided2.go
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: B
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided2.go
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided2.go
```

Deskripsi program

Pada program ini, saya membuat sebuah program untuk menentukan nilai mata kuliah (nmk) berdasarkan nilai akhir mata kuliah (nam) yang dimasukkan oleh pengguna. Variabel nam digunakan untuk menyimpan nilai akhir dalam bentuk float, sedangkan nmk digunakan untuk menyimpan nilai huruf seperti A, AB, B, BC, C, D, atau E.

Program dimulai dengan menampilkan pesan "Nilai akhir mata kuliah:" untuk meminta pengguna memasukkan nilai akhirnya. Nilai tersebut lalu dibaca menggunakan `fmt.Scan(&nam)`.

Setelah mendapatkan nilai akhir, program menggunakan beberapa kondisi `if` dan `else if` untuk menentukan nilai huruf. Setiap rentang nilai memiliki hasil huruf yang berbeda. Misalnya, jika nilai akhir lebih besar dari 80, maka nmk akan diisi dengan "A". Jika tidak memenuhi syarat tersebut, program melanjutkan memeriksa rentang berikutnya: lebih dari 72.5 untuk "AB", lebih dari 65 untuk "B", lebih dari 57.5 untuk "BC", lebih dari 50 untuk "C", dan lebih dari 40 untuk "D". Jika tidak ada satu pun kondisi yang terpenuhi, maka nilai huruf yang diberikan adalah "E".

Setelah nilai huruf berhasil ditentukan, program menampilkan "Nilai mata kuliah:" diikuti dengan isi dari variabel nmk.

Menjawab Pertanyaan di Modul 10 :

- 1.Jika NAM diberikan 80.1, maka program akan mengeksekusi kondisi yang pertama yaitu `if nam > 80` dia akan meng output NMK A
- 2.yang salah dalam program di modul 10 adalah :

- Setiap kondisi itu dituliskan if, seharusnya if hanya digunakan di kondisi pertama saja, setelahnya else if, dan yang terakhir jika perlu else, jika kita tulis if semua maka program akan membaca semua kondisi dan mengeksekusi kondisi yang benar dan posisi yang berada di akhir , jika kita gunakan if else if dan else, program akan mencari sampai bener dan jika sudah menemukan yang benar maka program langsung mengesekusi tanpa melanjutkan ke kondisi selanjutnya
- Seharusnya variabel yang ada didalam if yaitu nmk bukan nam, karena yang dibutuhkan itu nilai A, AB, B, BC, C, D, dan E

3. Tugas 3A

Source code

```
package main

import "fmt"

func main() {
    var b int

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

    fmt.Print("Faktor: ")
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Print(i, " ")
        }
    }
}
```

Screenshot program

The screenshot shows a terminal window with the following content:

```
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided3A.go
bilangan: 12
Faktor: 1 2 3 4 6 12
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10> go run Unguided3A.go
bilangan: 7
Faktor: 1
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Unguided IF Minggu 10>
```

Deskripsi program

Program 3a dibuat untuk menampilkan semua faktor dari sebuah bilangan bulat positif yang dimasukkan oleh pengguna. Pertama, program meminta input sebuah bilangan dan menyimpannya dalam variabel *b*. Setelah itu, program mencetak teks "Faktor:" sebagai penanda bahwa hasil berikutnya adalah daftar faktornya.

Untuk mencari faktor, program menggunakan perulangan dari 1 hingga *b*. Pada setiap iterasi, program memeriksa apakah nilai *i* membagi *b* tanpa sisa menggunakan kondisi *b % i == 0*. Jika kondisi terpenuhi, berarti *i* adalah faktor dari *b*, sehingga program mencetak nilai tersebut. Dengan cara ini, seluruh faktor bilangan dapat ditampilkan secara berurutan.

4. Tugas 3B

Source code

```
package main

import "fmt"

func main() {
    var b int
    var prima bool
    var JumlahFaktor int = 0
```

```
fmt.Println("bilangan: ")
fmt.Scan(&b)

fmt.Println("Faktor: ")
for i := 1; i <= b; i++ {
    if b%i == 0 {
        fmt.Println(i, " ")
        JumlahFaktor++
    }
}

fmt.Println()

if JumlahFaktor == 2 {
    prima = true
}

fmt.Println("prima: ", prima)

}
```

Screenshoot program

The screenshot shows a code editor interface with several tabs at the top: Welcome, Unguided1.go 3, Unguided2.go 1, Unguided3B.go 1 (which is the active tab), and Unguided3A.go 1. The main area displays the following Go code:

```
4
5 func main() {
6     var b int
7     var prima bool
8     var JumlahFaktor int = 0
9
10    fmt.Println("bilangan: ")
11    fmt.Scan(&b)
12
13    fmt.Println("Faktor: ")
14    for i := 1; i <= b; i++ {
15        if b%i == 0 {
16            fmt.Println(i, " ")
17            JumlahFaktor++
18        }
19    }
20
21    fmt.Println()
22
23    if JumlahFaktor == 2 {
24        prima = true
25    }
26
27    fmt.Println("prima: ", prima)
28
29
```

Below the code editor is a terminal window titled 'NAMA' showing the output of the program:

```
NAMA : RAFIF ARFA DHIA PRAJA
KELAS : S1IF-13-07
NIM : 109082500041
```

The terminal also shows the command used to run the program: go run Unguided3B.go.

Deskripsi program

Program 3b melanjutkan logika dari program 3a dengan menambahkan fitur untuk menentukan apakah bilangan tersebut merupakan bilangan prima. Perhitungan faktor dilakukan dengan cara yang sama seperti pada program sebelumnya, tetapi kali ini setiap kali sebuah faktor ditemukan, variabel JumlahFaktor akan ditambah satu. Setelah seluruh faktor ditemukan dan dicetak, program menghitung apakah jumlah faktor sama dengan dua. Sebuah bilangan hanya dikatakan prima jika memiliki tepat dua faktor, yaitu 1 dan dirinya sendiri. Jika kondisi JumlahFaktor == 2 terpenuhi, variabel prima diubah menjadi *true*. Terakhir, program menampilkan hasil pengecekan dengan mencetak nilai dari variabel prima.