

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 2

REVIEW STRUKTUR KONTROL



Oleh:

MUHAMMAD RUSDIYANTO

2311102053

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Program Main

Secara umum, program utama dalam Go memiliki kata kunci ``main``. Program utama, selalu memiliki 2 komponen, yaitu *package main* dan *func main()*. Program utama dalam Go akan digunakan sebagai *point of entry* atau program pertama yang akan dijalankan. Hal ini akan sangat berguna nantinya ketika keseluruhan program terdiri dari berbagai macam file yang berada di direktori berbeda.

Coding dalam Go

Coding dalam Go dilakukan layaknya bahasa pemrograman lain. Untuk Go, kode program disimpan dalam format `.go`. Kode program yang dibuat bisa dikompilasi menggunakan perintah ``go build file.go``. Perintah tersebut akan menghasilkan program `.exe`. Program juga bisa dijalankan tanpa membuat `.exe` dari kode, dengan menggunakan perintah ``go run file.go``. Dengan ini, test running program bisa dilakukan dengan lebih leluasa.

Tipe Data

Go memiliki 5 tipe data utama yaitu integer (untuk angka), real (untuk bilangan desimal/koma), boolean (true/false), karakter (1 huruf atau simbol), dan string (untuk kalimat). Tiap tipe data tersebut memiliki tipe data yang lebih spesifik, misalnya `int8` untuk integer berukuran 8 bit (-127 - 127), `uint8` untuk 8 bit integer bilangan positif (0 - 256), dan tipe data lain. Hal ini berguna dalam efisiensi memori, sehingga meningkatkan performa program dan mengurangi beban *device*.

Variabel

Ketika deklarasi variabel dalam Go, tipe data dari variabel harus disertakan baik secara eksplisit (`var nama string = "Adi"`) maupun implisit (`nama := "Adi"`). Nilai variabel bisa diambil dengan memanggil nama variabel. Alamat variabel bisa diakses dengan menambahkan tanda ``&`` di awal ketika memanggil nama variabel.

Nilai yang disimpan dalam variabel dapat diproses menggunakan operator aritmatika (+, -, *, /,...) dan operator logika (==, !=, <, >,...), tentunya dengan memperhatikan tipe data dari variabel terkait.

Jika nilai suatu variabel tidak berubah, maka ``const`` bisa digunakan untuk menggantikan ``var`` (`const PHI = 3.14`).

Input/Output

Package `"fmt"` digunakan untuk mengambil input dan menampilkan output. Untuk input, gunakan fungsi `fmt.Scan()` atau variasi sejenis. Sementara untuk output, gunakan fungsi `fmt.Print()` atau variasi sejenis. Variasi sejenis yang dimaksud

adalah variasi dari fungsi dengan output yang sama namun dengan beberapa fitur tambahan. Misalnya *fmt.Println()* otomatis menambahkan ``n`` di akhir output dan *fmt.Printf()* yang menampilkan output terformat (umumnya digunakan untuk menampilkan nilai dari variabel disamping teks output biasa).

Perulangan

Perulangan dalam Go menggunakan 1 *keyword* yaitu *for*. Dalam Go, *for* dapat digunakan layaknya *while* dan *repeat-until*, semacam 1 sintaks untuk semua jenis perulangan. Yang membedakan apakah looping terkait adalah *for*, *while*, atau *repeat-until* terletak dalam kondisinya. Perulangan akan terus dijalankan hingga kondisi *false*.

Looping dalam Go bisa secara otomatis mengetahui panjang suatu *array/slice* dengan menggunakan kata kunci *range*. Looping jug bisa dilakukan tanpa kondisi, dimana looping tersebut akan berjalan terus menerus (bisa dihentikan dengan menggunakan *keyword`break`*).

Percabangan

Percabangan dalam Go kurang lebih sama dengan deklarasi percabangan di bahasa pemrograman lain. Go memiliki 2 jenis percabangan, yaitu *if-else* dan *switch-case*. Kode dalam blok percabangan akan dijalankan ketika kondisi benar/*true*, jika tidak maka kode dalam blok tersebut akan di-*skip*.

II. GUIDED

2A – 1 | Code

```
package main

import "fmt"

func main() {
    var (
        satu, dua, tiga string
        temp            string
    )

    fmt.Print("Masukan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&tiga)
    fmt.Println("Output awal =", satu, dua, tiga)

    temp = satu
    satu = dua
    dua = tiga
    tiga = temp

    fmt.Println("Output akhir =", satu, dua, tiga)
}
```

2A – 1 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor1.go
Masukan input string: 1
Masukan input string: 2
Masukan input string: 3
Output awal = 1 2 3
Output akhir = 2 3 1
```

2A – 1 | Penjelasan

Program di atas merupakan sebuah program yang menerima 3 input berupa string dan mengatur posisinya sehingga input 1 berada di posisi 3 (akhir), input 2 di posisi 1 (awal), dan input 3 di posisi 2. Hal ini dapat dicapai dengan memindahkan nilai tiap variabel ke variabel lain dengan bantuan 1 variabel temporary untuk menyimpan nilai sementara.

2A – 2 | Code

```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Scanln(&tahun)
    fmt.Printf("Kabisat: %v", tahun%4 == 0)
}
```

2A – 2 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor2.go
2012
Kabisat: true

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor2.go
2011
Kabisat: false
```

2A – 2 | Penjelasan

Program di atas adalah program yang dapat menentukan tahun input adalah kabisat atau bukan. Hal ini ditentukan dengan menghitung sisa bagi dari tahun dengan 4 (tahun % 4). Jika sisa bagi adalah 0, maka tahun yang dimasukkan adalah tahun kabisat (true). Jika sisa bagi bukan 0, maka tahun yang dimasukkan bukan tahun kabisat (false).

2A – 3 | Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jejari float64
    fmt.Print("Jejari = ")
    fmt.Scanln(&jejari)
    fmt.Printf("\nBola dengan jejari %v memiliki volume %v
dan luas kulit %v", jejari, (4*math.Pi*math.Pow(jejari,
3))/3, 4*math.Pi*math.Pow(jejari, 2))
}
```

2A – 3 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor3.go
Jejari = 5
Bola dengan jejari 5 memiliki volume 523.5987755982989 dan luas kulit 314.1592653589793
```

2A – 3 | Penjelasan

Program di atas adalah program untuk menghitung volume dan luas permukaan dari suatu bola. Program ini menggunakan *package* “math” untuk membantu proses kalkulasi dengan menggunakan nilai Pi/Phi yang telah disediakan *package*, serta *function* `math.Pow()` untuk mengangkat bilangan. Tipe data input harus berupa float, supaya bisa digunakan untuk perhitungan. Perhitungan yang lain dilakukan seperti halnya menghitung volume dan luas dari suatu bola.

III. UNGUIDED

2A – 4 | Code

```
package main

import "fmt"

func main() {
    var celcius float64
    fmt.Println("Temperatur Celcius : ")
    fmt.Scanln(&celcius)
    fmt.Printf("Derajat Reamur : %v\n", (celcius * 4 / 5))
    fmt.Printf("Derajat Fahrenheir : %v\n",
(celcius*9/5)+32)
    fmt.Printf("Derajat Kelvin : %v\n", celcius+273.15)
}
```

2A – 4 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor4.go
Temperatur Celcius : 12
Derajat Reamur : 9.6
Derajat Fahrenheir : 53.6
Derajat Kelvin : 285.15
```

2A – 4 | Penjelasan

Program di atas merupakan program konversi suhu yang mengonversi suhu dari celcius ke derajat suhu lain (fahrenheit, reamur, dan kelvin). Perhitungan dilakukan dengan memproses input (celcius) lalu sesuai dengan rumus konversi celcius ke reamur, fahrenheit, dan kelvin secara berurut. Hasil perhitungan tidak disimpan ke dalam variabel, tetapi langsung di-outputkan.

2A – 5 | Code

```
package main

import "fmt"

func main() {
    var nums = [5]int{}
    var chars string

    fmt.Scanln(&nums[0], &nums[1], &nums[2], &nums[3],
&nums[4])
    fmt.Scanln(&chars)
```

```

    for i := 0; i < 5; i++ {
        fmt.Printf("%c", nums[i])
    }
    fmt.Print("\n")
    for i := 0; i < 3; i++ {
        fmt.Printf("%c", byte(chars[i])+1)
    }
}

```

2A – 5 | Output

```

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor5.go
65 69 110 53 72
VHG
AEn5H
WIH

```

2A – 5 | Penjelasan

Program di atas merupakan program yang mengonversi angka menjadi karakter (berdasarkan tabel ASCII/UTF-8) dan menampilkan huruf selanjutnya berdasarkan 3 karakter yang dimasukkan. Proses dimulai dengan memasukkan 5 angka, disusul dengan 3 karakter. 5 angka akan disimpan dalam array `nums`, sementara 3 karakter akan disimpan dalam string `chars`. Setelah itu, program akan melakukan perulangan untuk setiap angka dalam `nums`, dimana program akan mengoutputkan karakter dalam tabel ASCII yang nilainya sama dengan nilai integer pada indeks `i`. Setelah itu, program akan kembali melakukan perulangan untuk setiap karakter dalam `chars`. Untuk proses outputnya, `chars` akan di-*typecast* menjadi *byte array*. Byte dalam Go adalah tipe data untuk menyimpan karakter ASCII/UTF-8, sehingga cocok untuk keperluan ini. Setelah tipe data diubah, maka nilai integer karakter akan ditambah 1, lalu nilai yang telah ditambahkan tersebut akan ditampilkan sebagai karakter lagi. Proses tersebut akan menampilkan karakter berikutnya dari karakter yang dimasukkan.

2B – 1 | Code

```

package main

import (
    "bufio"
    "fmt"
    "os"
)

```



```

func main() {
    var result, prevResult string
    var conclusion bool = true
    input := bufio.NewScanner(os.Stdin)

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %v : ", i)
        input.Scan()
        result = input.Text()
        if i == 1 {
            prevResult = result
            continue
        }
        if result != prevResult {
            conclusion = false
        }
    }
    fmt.Printf("BERHASIL : %v", conclusion)
}

```

2B – 1 | Output

```

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor1b.go
Percobaan 1 : merah kuning hijau biru
Percobaan 2 : merah kuning hijau biru
Percobaan 3 : merah kuning hijau biru
Percobaan 4 : biru kuning hijau merah
Percobaan 5 : merah kuning hijau biru
BERHASIL : false

```

```

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor1b.go
Percobaan 1 : merah kuning hijau biru
Percobaan 2 : merah kuning hijau biru
Percobaan 3 : merah kuning hijau biru
Percobaan 4 : merah kuning hijau biru
Percobaan 5 : merah kuning hijau biru
BERHASIL : true

```

2B – 1 | Penjelasan

Program di atas adalah program yang menentukan sukses atau tidaknya suatu percobaan berdasarkan sama atau tidaknya percobaan yang dilakukan. Jika salah satu percobaan memberikan hasil yang tidak sama, maka percobaan akan dianggap gagal. Program ini menggunakan 2 package tambahan yaitu “bufio” dan “os”. Secara sederhana, kedua package tersebut digunakan supaya program dapat menerima input string yang disertai dengan spasi/whitespace. Proses dimulai dengan deklarasi variabel serta scanner. Scanner disini berfungsi untuk menerima input, layaknya fungsi

fmt.Scan(). Setelah itu, program akan melakukan perulangan sebanyak 5 kali. Di dalam perulangan, program akan menerima input (dari perintah *input.Scan()*). Setelah input selesai, maka nilai input akan disimpan di dalam variabel dengan memanggil fungsi *input.Text()*. Fungsi tersebut berguna untuk mendapatkan value yang telah di-scan oleh scanner saat proses input. Setelah itu program akan mengecek apakah iterasi sekarang ada dalam iterasi pertama. Jika iya, maka program akan menyimpan hasil percobaan dalam variabel ``prevResult`` dan langsung melanjutkan iterasi tanpa menjalankan kode program di bawahnya. Jika tidak, maka program akan kembali mengecek, apakah hasil iterasi sekarang berbeda dengan hasil iterasi pertama. Jika iya, maka program akan mengubah nilai conclusion menjadi *false*, menandakan bahwa percobaan gagal. Setelah 5 kali looping, maka program akan menampilkan berhasil atau tidaknya percobaan berdasarkan nilai yang disimpan dalam variabel ``conclusion``.

2B – 2 | Code

```
package main

import "fmt"

func main() {
    var flower, flowers string
    var flowerCount int = 0

    for true {
        fmt.Printf("Bunga %v : ", flowerCount+1)
        fmt.Scanln(&flower)
        if flower == "SELESAI" {
            break
        }
        flowers += flower + " - "
        flowerCount++
    }
    fmt.Printf("Pita : %v\nBunga : %v", flowers,
flowerCount)
}
```

2B – 2 | Output

```

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor2b.go
Bunga 1 : Kertas
Bunga 2 : Tulip
Bunga 3 : Mawar
Bunga 4 : SELESAI
Pita : Kertas - Tulip - Mawar -
Bunga : 3

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor2b.go
Bunga 1 : SELESAI
Pita :
Bunga : 0

```

2B – 2 | Penjelasan

Program di atas merupakan program input list bunga untuk sebuah pita. Program ini berjalan secara terus menerus. Dalam kata lain, program akan terus menerima input nama bunga sampai tak hingga. Di dalam perulangan, program pertama – tama akan menerima input, dimana input itu akan dicek. Jika input sama dengan ‘SELESAI’, maka program akan memberhentikan looping dan mengoutputkan daftar nama bunga dan jumlah bunga yang di-inputkan. Jika tidak, maka program akan menambahkan nama bunga disertai dengan tanda setrip “ - ” dan menambahkan jumlah bunga dengan 1.

2B – 3 | Code

```

package main

import "fmt"

func main() {
    for true {
        var kantongKanan, kantongKiri float64
        fmt.Print("Masukkan berat belanjaan kantong : ")
        fmt.Scanln(&kantongKanan, &kantongKiri)
        if kantongKanan+kantongKiri > 150 {
            fmt.Println("Proses selesai.")
            break
        }
        fmt.Printf("Sepeda motor Pak Andi akan
oleng : %v\n", kantongKanan-kantongKiri >= 9 ||
kantongKanan-kantongKiri <= -9)
    }
}

```

2B – 3 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor3b.go
Masukkan berat belanjaan kantong : 30 40
Sepeda motor Pak Andi akan oleng : true
Masukkan berat belanjaan kantong : 5 11
Sepeda motor Pak Andi akan oleng : false
Masukkan berat belanjaan kantong : 120 35
Proses selesai.
```

2B – 3 | Penjelasan

Program di atas merupakan program yang mengecek apakah motor Pak Andi akan oleng berdasarkan berat barang yang akan dia bawa di kedua sisi motornya. Program dimulai dengan sebuah perulangan yang tidak akan pernah berhenti. Di dalam perulangan tersebut, proses akan dimulai dengan input dua berat, kanan dan kiri. Setelah itu, program akan mengecek apakah jumlah dari kedua berat lebih dari 150 kg. Jika iya, maka perulangan akan diberhentikan menggunakan *keyword* `break`. Jika tidak, maka program akan lanjut dengan mengecek apakah selisih berat kanan dan kiri lebih dari sama dengan 9 sekaligus menampilkan hasil pengecekan tersebut.

2B – 4 | Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k float64
    var result float64 = 1
    fmt.Print("Nilai K = ")
    fmt.Scanln(&k)
    for i := 0; i <= int(k); i++ {
        result *= math.Pow(4*float64(i)+2, 2) /
float64(((4*i + 1) * (4*i + 3)))
    }
    fmt.Printf("Nilai akar 2 = %v\n", result)
}
```

2B – 4 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor4b.go
Nilai K = 100
Nilai akar 2 = 1.41333870718477
```

2B – 4 | Penjelasan

Program tersebut adalah program yang menghitung kisaran nilai dari akar 2. Akar 2 merupakan bilangan irrasional sehingga sulit untuk mencari nilainya. Program ini mencari nilai tersebut dengan bantuan looping. Semakin banyak iterasi yang dilakukan oleh program, maka nilai akar 2 akan semakin mendekati nilai aslinya. Proses pencarian nilai dimulai dengan input berapa banyak iterasi yang akan dilakukan. Setelah itu, program akan melakukan perulangan, dimana di dalam perulangan tersebut, terdapat rumus untuk mencari hasil pada iterasi tersebut. Nantinya, hasil akan dikalikan secara terus menerus dengan variabel `result`. Di dalam rumus, terdapat variabel `i` yang nilainya akan selalu berubah setiap iterasi. Setelah iterasi selesai, maka program akan mengoutputkan hasil kisaran nilai dari akar 2.

2C – 1 | Code

```
package main

import "fmt"

func main() {
    var berat, sisaBerat, biayaBerat, biayaSisaBerat, total
    int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scanln(&berat)
    sisaBerat = berat % 1000
    berat -= sisaBerat
    biayaBerat = berat * 10
    fmt.Printf("Detail berat : %v kg + %v gr\n", berat/1000,
sisaBerat)
    if sisaBerat >= 500 {
        biayaSisaBerat = sisaBerat * 5
    } else {
        biayaSisaBerat = sisaBerat * 15
    }
    fmt.Printf("Detail biaya : Rp. %v + Rp. %v\n",
biayaBerat, biayaSisaBerat)
    if berat > 10000 {
        total = biayaBerat
    } else {
        total = biayaBerat + biayaSisaBerat
    }
    fmt.Printf("Total biaya : Rp. %v", total)
}
```

2C – 1 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor1c.go
Berat parcel (gram): 8750
Detail berat : 8 kg + 750 gr
Detail biaya : Rp. 80000 + Rp. 3750
Total biaya : Rp. 83750

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor1c.go
Berat parcel (gram): 8250
Detail berat : 8 kg + 250 gr
Detail biaya : Rp. 80000 + Rp. 3750
Total biaya : Rp. 83750

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor1c.go
Berat parcel (gram): 11750
Detail berat : 11 kg + 750 gr
Detail biaya : Rp. 110000 + Rp. 3750
Total biaya : Rp. 110000
```

2C – 1 | Penjelasan

Program tersebut merupakan program untuk menghitung total biaya pengiriman suatu parcel. Pertama program akan menerima input berupa berat parcel dalam gram. Lalu, berat tersebut akan dibagi dengan 1000 lalu diambil sisa baginya. Sisa bagi tersebut adalah sisa berat yang nantinya akan dikalkulas secara independen. Berat utama parcel dikurangi dengan sisa berat parcel, lalu biaya berat utama parcel bisa didapatkan dengan dikalikan 10 (1 kg = Rp. 10.000, 1 g = Rp. 10). Berat dan sisa berat di outputkan, lalu untuk kalkulasi sisa berat parcel akan terbagi menjadi 3. Jika sisa berat lebih dari 500 gr, maka sisa berat akan dikalikan dengan 5. Jika sisa berat kurang dari 500 gr, maka sisa berat akan dikalikan dengan 15. Untuk sementara, biaya utama dan biaya sisa berat akan di output untuk memperlihatkan biaya sebelum ditotalkan. Sebelum ditotalkan, program akan mengecek apakah berat lebih besar dari 10.000 (10 kg). Jika iya, maka total akan sama dengan biaya utama saja (biaya sisa berat diabaikan). Jika tidak, maka total akan sama dengan biaya utama ditambah dengan biaya sisa berat. Setelah itu, total biaya akan ditampilkan.

2C – 2 | Code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah : ")
}
```

```

fmt.Scanln(&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.Printf("Nilai mata kuliah : %v", nmk)
}

```

2C – 2 | Output

```

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor2c.go
Nilai akhir mata kuliah : 93.5
Nilai mata kuliah : A
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor2c.go
Nilai akhir mata kuliah : 70.6
Nilai mata kuliah : B
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor2c.go
Nilai akhir mata kuliah : 49.5
Nilai mata kuliah : D

```

2C – 2 | Penjelasan

a.) Berdasarkan kode yang diberikan, program tidak bisa dijalankan (error) karena `nam` tidak bisa menyimpan tipe data string. Dengan asumsi bahwa hal tersebut merupakan kesalahan dari pembuat soal, maka program bisa dijalankan. Meskipun begitu, jika memasukkan `nam = 80.1`, keluaran program adalah ``D``. Hal ini belum sesuai dengan spesifikasi soal. Seharusnya, program mengeluarkan output ``A``. Karena `80.1` lebih besar dari `80`. Hal ini dikarenakan percabangan tidak menggunakan ``else-if`` untuk semua kategori nilai, sehingga ketika salah satu kondisi nilai bernilai `true` (contohnya `80.1 > 80`), maka program akan mengecek nilainya lagi, walaupun seharusnya sudah tidak perlu. Akhirnya nilai dari akan lebih rendah dari nilai input.

b.) Kesalahan kode terletak dalam setiap percabangan. Seharusnya semua percabangan disatukan menjadi deretan percabangan `if-else` dan variabel yang akan diberikan nilai seharusnya adalah `nmk`, bukan `nam`.

c.) [terlampir di bagian output]

2C – 3 | Code

```
package main

import "fmt"

func main() {
    var num int
    var isPrime bool = true
    fmt.Print("Bilangan : ")
    fmt.Scanln(&num)
    fmt.Print("Faktor : ")
    for i := 1; i <= num; i++ {
        if num%i == 0 {
            fmt.Printf("%v ", i)
            if i != 1 && i != num {
                isPrime = false
            }
        }
    }
    if num < 2 {
        isPrime = false
    }
    fmt.Printf("\nPrima : %v", isPrime)
}
```

2C – 3 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor3c.go
Bilangan : 8
Faktor : 1 2 4 8
Prima : false

PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day2> go run nomor3c.go
Bilangan : 13
Faktor : 1 13
Prima : true
```

2C – 3 | Penjelasan

Program di atas adalah program penghitung faktor suatu bilangan sekaligus pengecek apakah bilangan tersebut adalah bilangan prima. Cara kerja program yang satu ini adalah dengan melakukan perulangan dari angka satu

sampai dengan angka yang dimasukan. Di dalam perulangan akan dilakukan pengecekan apakah nilai angka di perulangan tersebut bisa habis membagi angka yang dimasukkan. Jika iya, maka angka akan di outputkan dan dicek lagi, apakah angka terkait bukan 1 dan bukan angka input. Hal ini dilakukan unuk mengecek apakah bilangan tersebut merupakan bilangan prima. Karena bilangan prima hanya bisa habis dibagi oleh 1 dan angka itu sendiri. Jika iya, maka program akan mengubah nilai ``isPrime`` menjadi false. Jika tidak, maka program akan *skip* ke iterasi selanjutnya. Setelah perulangan selesai, maka akan kembali mengecek apakah nilai angka kurang dari 2. Hal ini dilakukan jikalau angka yang dimasukkan adalah 1, dimana dalam konteks ini, 1 bukanlah bilangan prima. Jika ternyata nilai angka kurang dari 2, maka program akan mengubah nilai ``isPrime`` menjadi false. Terakhir, program akan memberikan output terkait prima atau tidaknya angka inputan.