

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 10**  
**“PERCABANGAN”**



**DISUSUN OLEH:**  
**DWI OKTA SURYANINGRUM**  
**103112400066**  
**S1 IF-12-01**  
**DOSEN:**  
**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024/2025**

## **DASAR TEORI**

### **PERCABANGAN**

#### **1. Pernyataan If**

Dalam Bahasa, if berarti Jika, contoh pernyataan: Jika hujan maka mandi hujan. Pernyataan Jika digunakan untuk kita memilih kondisi berdasarkan benar atau tidak, jika benar dianggap memenuhi kondisi dan sebaliknya.

#### **2. Pernyataan If-Else**

Bila kondisi tidak memenuhi pernyataan If maka blok kode akan dilewat, bagian ini membahas blok kode bila tidak memenuhi pernyataan maka akan dijalankan : Pernyataan else (selain).

#### **3. Pernyataan Else-if**

Pernyataan ini bertujuan untuk menyambungkan antar pernyataan if. else-if dalam penerapan ini digunakan untuk menguji batasan yang lebih diutamakan dalam suatu kasus.

#### **4. Pernyataan switch**

Switch bila diartikan ke kata benda dalam bahasa : saklar, filosofinya bila ada nilai yang sesuai dengan kasus saklar (case) maka akan menjalankan blok kode didalamnya seperti pernyataan if. Perbedaanya, pernyataan switch menguji nilai apakah cocok tidak hanya mengecek data logik dari operasi (data numerik dan karakter juga dicek).

Percabangan secara singkat, memberikan alur bagi kode kita dengan memberikan saklar kondisi pada bagian kode kita untuk dijalankan. Batasan kondisi yang menjadi basis percabangan untuk bekerja mengecek nilai variable / hasil operasi yang menjadi saklar untuk blok mana yang akan dieksekusi. Jadi, Penerapan percabangan menjadi pengatur rute berjalan program (logika program) agar program memiliki karakter sehingga tidak hanya menjalankan prosedur terurut dan monoton.

## CONTOH SOAL

### 1. Latihan1

Buatlah program dengan bahasa Go yang digunakan untuk menentukan apakah seorang penduduk bisa membuat KTP atau tidak. Syarat utama membuat KTP adalah berusia minimal 17 tahun dan memiliki kartu keluarga.

**Masukan** terdiri dari dua baris. Baris pertama adalah bilangan bulat yang menyatakan usia, sedangkan baris kedua adalah boolean yang menyatakan memiliki KTP atau tidak.

**Keluaran** berupa teks "bisa membuat KTP" atau "belum bisa membuat KTP" sesuai dengan masukan yang diberikan.

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1.	17 True	Bisa buat KTP
2.	20 False	Belum bisa buat KTP
3.	15 True	Belum bisa buat 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")
    }
}

```

Output:

```

▼ TERMINAL
● mymac Terminal MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1/Modul10/coso/coso1.go"
17 true
Bisa membuat KTP
○ mymac@mymacs-MacBook-Air praktikum1 %
○ mymac@mymacs-MacBook-Air praktikum1 %
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1/Modul10/coso/coso1.go"
20 false
Belum bisa membuat KTP
○ mymac@mymacs-MacBook-Air praktikum1 %
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1/Modul10/coso/coso1.go"
15 true
Belum bisa membuat KTP

```

Deskripsi Program:

- var usia mendeklarasikan variabel dengan tipe data integer, var kk mendeklarasikan variabel dengan tipe data boolean
- fmt.Scan(&usia, &kk) digunakan untuk menginputkan nilai var usia dan kk
- if usia >= 17 && kk merupakan sebuah kondisi yang menyatakan jika usia memiliki nilai lebih dari atau sama dengan 17 dan mempunyai kk (kk bernilai true) , maka outputnya adalah “bisa membuat KTP”
- namun jika salah satu kondisi tidak terpenuhi maka outputnya adalah “belum bisa membuat KTP”

## 2. Latihan 2

Buatlah suatu program Go yang digunakan untuk menentukan suatu alfabet yang diberikan adalah vokal atau konsonan.

**Masukan** terdiri dari suatu huruf.

**Keluaran** berupa teks yang menyatakan huruf adalah "vokal" atau "konsonan" atau "bukan huruf".

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1.	A	Vocal
2.	f	Konsonan
3.	1	Bukan huruf
4.	\$	Bukan huruf

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")
    }
}
```

Output:

```
✓ TERMINAL
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
  /Modul10/coso/coso2.go"
  A
  vokal
○ mymac@mymacs-MacBook-Air praktikum1 %
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
  /Modul10/coso/coso2.go"
  f
  konsonan
○ mymac@mymacs-MacBook-Air praktikum1 %
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
  /Modul10/coso/coso2.go"
  1
  bukan huruf
○ mymac@mymacs-MacBook-Air praktikum1 %
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
  /Modul10/coso/coso2.go"
  $
  bukan huruf
```

Deskripsi Program:

- var x mendeklarasikan sebuah variabel dengan tipe rune, var huruf, vkecil, vbesar mendeklarasikan 3 variabel dengan tipe data Boolean
- `fmt.Scanf("%c", &x)` digunakan untuk membaca karakter tunggal
- Pada var huruf memeriksa apakah x adalah huruf (baik kecil atau besar).
- Pada var vkecil memeriksa apakah x adalah vokal kecil ('a', 'i', 'u', 'e', 'o')
- Pada var vbesar memeriksa apakah x adalah vokal kecil ('A', 'I', 'U', 'E', 'O')
- Pada percabangan dijelaskan output berupa vokal jika x merupakan huruf dan vokal.
- output berupa konsonan jika x adalah huruf dan bukan vokal.
- output berupa bukan huruf jika x bukan huruf.

### 3. Latihan 3

Buatlah program untuk menentukan apakah digit dalam suatu bilangan terurut membesar, mengecil, atau tidak terurut. Bilangan hanya terdiri dari empat digit saja, atau lebih besar atau sama dengan 1000 dan kecil sama atau sama dengan 9999.

Perhatikan contoh sesi interaksi program berikut (teks bergaris bawah adalah input/read).

**Contoh masukan dan keluaran:**

No	Masukan dan Keluaran
1.	Bilangan : 2489 Digit pada baingan 2489 terurut membesar
2.	Bilangan : 3861 Digit pada baingan 3861 tidak terurut
3.	Bilangan : 9651 Digit pada baingan 9651 terurut mengecil

Source Code

```
package main
import "fmt"
func main() {
    var bilangan, d1,d2,d3,d4 int
    var teks string
    fmt.Print("Bilangan : ")
    fmt.Scan(&bilangan)
    d1 = bilangan / 1000
    d2 = (bilangan%1000) / 100
    d3 = (bilangan%100) / 10
    d4 = bilangan % 10
    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)
}
```

Output:

## ✓ TERMINAL

```
● mymac@mymacs-MacBook-Air praktikum1 % go run
"/Users/mymac/praktikum1/Modul10/coso/tempCode
RunnerFile.go"
Bilangan : 2489
Digit pada bilangan 2489 Terurut membesar
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
/Modul10/coso/tempCodeRunnerFile.go"
Bilangan : 3861
Digit pada bilangan 3861 Tidak terurut
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
/Modul10/coso/tempCodeRunnerFile.go"
Bilangan : 9651
Digit pada bilangan 9651 Terurut mengecil
○ mymac@mymacs-MacBook-Air praktikum1 %
```

### Deskripsi Program:

- Var bilangan, d1,d2,d3,d4 mendeklarasikan 5 variabel dengan tipe data integer, var teks mendeklarasikan sebuah variabel dengan tipe data string
- `fmt.Scan(&bilangan)` digunakan untuk menginputkan nilai dari var bilangan
- d4: Mendapatkan digit terakhir (satuan) dari bilangan dengan menggunakan operator modulo `% 10`.
- d3: Mendapatkan digit kedua dari belakang (puluhan) dengan cara mengambil sisa bagi bilangan dibagi 100, lalu dibagi 10.
- d2: Mendapatkan digit ketiga dari belakang (ratusan) dengan cara mengambil sisa bagi bilangan dibagi 1000, lalu dibagi 100.
- d1: Mendapatkan digit pertama (ribuan) dengan membagi bilangan dengan 1000.
- **if d1 < d2 && d2 < d3 && d3 < d4**: Memeriksa apakah digit-digit angka (d1, d2, d3, d4) terurut membesar (dari kiri ke kanan semakin besar).
- Jika benar, maka teks di-set menjadi "terurut membesar".
- **else if d1 > d2 && d2 > d3 && d3 > d4**: Memeriksa apakah digit-digit angka terurut mengecil (dari kiri ke kanan semakin kecil).
- Jika benar, maka teks di-set menjadi "terurut mengecil".
- **else**: Jika kedua kondisi sebelumnya tidak terpenuhi, maka digit tidak terurut dengan jelas, sehingga teks di-set menjadi "tidak terurut".



## SOAL LATIHAN

### Statement Percabangan

#### 1. Latihan Soal 1

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel.

Maka, buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!

Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

No	Masukan dan Keluaran
1.	Berat parcel (gram): 8500 Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2.	Berat parcel (gram): 9250 Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3.	Berat parcel (gram): 11750 Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 110000

Source Code:

```
package main
import "fmt"
func main() {
```

```

        var berat_parsel, detail_biaya, total_berat, sisa_berat,
sisa int
        fmt.Print("Berat parsel (gram) : ")
        fmt.Scan(&berat_parsel)
        total_berat = berat_parsel/1000
        sisa_berat = berat_parsel%1000
        fmt.Printf("Detail Berat : %d kg + %d gr\n", total_berat,
sisa_berat)
        detail_biaya = total_berat*10000
        if sisa_berat >= 500 {
            sisa = sisa_berat*5
        }else if sisa_berat < 500{
            sisa = sisa_berat*15
        }else if sisa_berat>10{
            sisa = 0
        }
        fmt.Printf("Detail biaya: Rp. %d + Rp. %d \n",
detail_biaya, sisa)
        fmt.Printf("Total biaya: Rp. %d \n", detail_biaya+sisa)
    }

```

Output:

```

▼ TERMINAL
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
/Modul10/latsol/latsol1.go"
Berat parsel (gram) : 8500
Detail Berat : 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
/Modul10/latsol/latsol1.go"
Berat parsel (gram) : 9250
Detail Berat : 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
/Modul10/latsol/latsol1.go"
Berat parsel (gram) : 11750
Detail Berat : 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 113750

```

#### Deskripsi Program:

- var berat\_parsel, detail\_biaya, total\_berat, sisa\_berat, sisa mendeklarasikan 5 variabel dengan tipe data integer
- `fmt.Scan(&berat_parsel)` digunakan untuk menginputkan nilai dari var berat\_parsel
- `total_berat = berat_parsel / 1000` artinya menghitung total berat dalam satuan kilogram dengan membagi berat parsel dalam gram (berat\_parsel) dengan 1000.
- `sisa_berat = berat_parsel % 1000` maksudnya menghitung sisa berat dalam gram yang tidak dapat dikonversi ke kilogram, dengan menggunakan operasi modulus (%).
- Kemudian `fmt.Printf()` menampilkan hasil konversi berat dalam format "kg + gr", dengan total\_berat untuk kilogram dan sisa\_berat untuk gram.
- Menghitung biaya pengiriman berdasarkan berat total dalam kilogram dalam variabel detail\_biaya dengan biaya pengiriman adalah 10.000 rupiah per kilogram, jadi `total_berat * 10000` memberikan total biaya untuk berat dalam kilogram.
- **if sisa\_berat >= 500** menyatakan kondisi dimana jika sisa berat lebih dari atau sama dengan 500 gram, maka biaya tambahan dihitung sebagai `sisa_berat * 5`.
- **else if sisa\_berat < 500** menyatakan kondisi jika sisa berat kurang dari 500 gram, maka biaya tambahan dihitung sebagai `sisa_berat * 15`.
- **else if sisa\_berat > 10** menyatakan kondisi jika `sisa_berat > 10(kg)` maka tidak ada biaya tambahan atau **sisa = 0**

## 2. Latihan Soal 2

Buatlah Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

No	NAM	NMK
1.	$NAM > 80$	A
2.	$72.5 < NAM \leq 80$	AB

3.	$65 < \text{NAM} \leq 72.5$	B
4.	$57.5 < \text{NAM} \leq 65$	BC
2.	$50 < \text{NAM} \leq 57.5$	C
3.	$40 < \text{NAM} \leq 50$	D
4.	$\text{NAM} \leq 40$	E

Program berikut menerima input sebuah bilangan riil yang menyatakan NAM.  
Program menghitung NMK dan menampilkannya.

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 {
        nam = "A"
    } if nam > 72.5 {
        nam = "AB"
    } if nam > 65 {
        nam = "B"
    } if nam > 57.5 {
        nam = "BC"
    } if nam > 50 {
        nam = "C"
    } if nam > 40 {
        nam = "D"
    } else if nam <= 40 {
        nam = "E"
    }
}
```

```

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

```

Output:

```

✓ TERMINAL
ⓧ mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1
/Modul10/latsol/latsol2.go"
# command-line-arguments
Modul10/latsol/latsol2.go:7:6: declared and not used: nmk
Modul10/latsol/latsol2.go:11:9: cannot use "A" (untyped string consta
nt) as float64 value in assignment
Modul10/latsol/latsol2.go:14:9: cannot use "AB" (untyped string const
ant) as float64 value in assignment
Modul10/latsol/latsol2.go:17:9: cannot use "B" (untyped string consta
nt) as float64 value in assignment
Modul10/latsol/latsol2.go:20:9: cannot use "BC" (untyped string const
ant) as float64 value in assignment
Modul10/latsol/latsol2.go:23:9: cannot use "C" (untyped string consta
nt) as float64 value in assignment
Modul10/latsol/latsol2.go:26:9: cannot use "D" (untyped string consta
nt) as float64 value in assignment
Modul10/latsol/latsol2.go:28:9: cannot use "E" (untyped string consta
nt) as float64 value in assignment
○ mymac@mymacs-MacBook-Air praktikum1 % █

```

Jawablah pertanyaan-pertanyaan berikut:

- Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
- Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
- Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

### Jawaban

- Keluaran program jika diberikan nilai 80.1 adalah "cannot use "A" (untyped string constant) as float64 value in assignment" artinya program ini tidak akan menghasilkan output karena terdapat kesalahan tipe data. Sehingga eksekusi program asli tidak sesuai dengan spesifikasi soal, karena terjadi kesalahan tipe data dan kesalahan dalam urutan pengkondisian.
- Kesalahan :

- Variabel nam bertipe float64, namun diubah menjadi string di dalam kondisi if, yang tidak sesuai dengan tipe data float64.
- Perbaikan : menggunakan var nmk untuk menyimpan nilai huruf (string).
- Struktur if saling bertumpuk. Program mengevaluasi setiap kondisi meskipun kondisi sebelumnya sudah terpenuhi.
- Perbaikan : menggunakan else if untuk memastikan hanya satu kondisi yang dieksekusi, dan atur rentang nilai yang benar.

c. Kode yang sudah diperbaiki

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 if nam <= 40 {

        nmk = "E"

    }

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

}

```

### Output

```

● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1/Modul10/latsol/latsol2.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1/Modul10/latsol/latsol2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1/Modul10/latsol/latsol2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
○ mymac@mymacs-MacBook-Air praktikum1 % 

```

### 3. Latihan Soal 3

Sebuah bilangan bulat  $b$  memiliki faktor bilangan  $f > 0$  jika  $f$  habis membagi  $b$ . Contoh: 2 merupakan faktor dari bilangan 6 karena 6 habis dibagi 2. Buatlah program yang menerima input sebuah bilangan bulat  $b$  dan  $b > 1$ . Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut! Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bilangan : 12	Bilangan : 7
Faktor : 1,2,3,4,6,12	Faktor : 1,7

Bilangan bulat  $b > 0$  merupakan bilangan prima  $p$  jika dan hanya jika memiliki persis dua faktor bilangan saja, yaitu 1 dan dirinya sendiri. Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat  $b > 0$ . Program tersebut mencari

dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah b merupakan bilangan prima. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bilangan : 12	Bilangan : 7
Faktor : 1,2,3,4,6,12	Faktor : 1,7
Prima = false	Prima : true

#### Source Code

```
package main
import "fmt"
func main() {
    var b int
    fmt.Scan(&b)
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Print(i, " ")
        }
    }

    if b%2 == 0 || b%3 == 0 || b%5 == 0 || b%7 == 0 && b != 1
    && b != 2 && b != 3 && b != 5 && b != 7 {
        fmt.Println("\nFalse")
    } else {
        fmt.Println("\nTrue")
    }
}
```

Output:



```

● mymac@mymacs-MacBook-Air praktikum1 % go run "
/Users/mymac/praktikum1/Modul10/latsol/tempCode
RunnerFile.go"
12
1 2 3 4 6 12
False
● mymac@mymacs-MacBook-Air praktikum1 % go run "/Users/mymac/praktikum1/
Modul10/latsol/tempCodeRunnerFile.go"
7
1 7
True
○ mymac@mymacs-MacBook-Air praktikum1 %

```

#### Deskripsi Program :

- var b mendeklarasikan sebuah variabel dengan tipe data integer
- **for i := 1; i <= b; i++** melakukan perulangan dari i = 1 hingga i = b untuk mencari faktor-faktor dari bilangan b
- **if b%i == 0** merupakan sebuah kondisi yang memeriksa apakah i adalah faktor dari b. Jika benar, maka i akan dicetak.
- **fmt.Print(i, " ")** akan mencetak setiap faktor i yang ditemukan, dipisahkan dengan spasi.
- **if b%2 == 0 || b%3 == 0 || b%5 == 0 || b%7 == 0 && b != 1 && b != 2 && b != 3 && b != 5 && b != 7** memeriksa apakah bilangan b bisa dibagi oleh 2, 3, 5, atau 7, namun b tidak boleh sama dengan 1, 2, 3, 5, atau 7.
- Jika salah satu kondisi benar, maka program mencetak "False", yang berarti bilangan b bukan bilangan prima.
- **else { fmt.Println("\nTrue") }** Jika kondisi di atas tidak terpenuhi, maka program mencetak "True", yang berarti b adalah bilangan prima.

## DAFTAR PUSTAKA

Ywnv. 2024. "Golang X04 - Kondisi Percabangan." *Baru Ngoding*. Accessed November 16.  
<https://barungoding.vercel.app/t/go-percabangan/>.