

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 13**  
**“REPEAT UNTIL”**



**DISUSUN OLEH:**  
**RYAN AKEYLA NOVIANTO WIDODO**  
**103112400081**  
**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(DEFAULT)**

### **1.Data**

Pengertian data adalah Data: Hasil pencatatan penelitian berupa fakta atau angka.

### **2.Tipe Data**

Dalam pemrograman, tipe data adalah jenis nilai yang akan disimpan dalam variabel. Tipe data berfungsi untuk memberi tahu sistem komputer bagaimana menafsirkan nilai data. Ada 5 tipe Data.

Tipe-Tipe Data:

- 1.Integer, seperti: int, biasanya digunakan pada bilangan bulat
- 2.real, seperti: float64, biasanya digunakan pada bilangan decimal
3. boolean (atau logikal), seperti: bool, biasanya digunakan pada true false
- 4.karakter, seperti: byte, biasanya digunakan pada nama suatu benda atau apapun
- 5.string, seperti: string, biasanya digunakan pada suatu nama benda dan bisa menggunakan angka, hampir sama seperti karakter

### **3.Variabel**

Variabel: Simbol atau nama yang digunakan untuk menyimpan nilai data dalam memori komputer. Variabel dapat berubah nilainya selama program komputer dijalankan.

Variabel adalah nama yang mewakili suatu elemen data, seperti tanggal lahir, untuk tempat lahir, alamat untuk alamat, tangla untuk tanggal lahir, dsb. Ada aturan tertentu yang wajib diikuti dalam pemberian nama variabel, antara lain:

1. Harus dimulai dengan abjad, tidak boleh dimulai dengan angka atau symbol. Khusus untuk PHP, variabel selalu ditulis dengan awalan berupa &.
2. Tidak boleh ada spasi.
3. Jangan menggunakan symbol-simbol yang bisa membingungkan, seperti titik dua, titik koma, koma, dsb.
4. Sebaiknya memiliki arti yang sesuai dengan elemen data.
5. Sebaiknya tidak terlalu panjang. Contoh nama variabel yang benar &nama, &nilai\_ujian  
Contoh nama variabel yang salah 4xyz, &ip rata.

Variabel menurut saya adalah suatu simbol yang bisa diibaratkan seperti sebuah wadah. Apa maksud wadah itu? Maksudnya adalah wadah itu biasa diisi suatu benda. Jadi variabel itu bisa diibaratkan seperti: mobil, computer, motor, handphone, pohon, ataupun benda yang lainnya. Contohnya: suatu x merupakan benda yang bisa dimakan dan y adalah minuman. Nah di sinilah variabel diibaratkan seperti itu. Jadi x bisa menjadi nasi goreng, atau apapun makanan, dan y bisa menjadi minuman apapun seperti: Es teh, Es jeruk, dll.

#### 4. Deklarasi Variabel

Deklarasi variabel memastikan program memiliki informasi yang cukup tentang variabel sebelum digunakan, yang mana akan membantu dalam menghindari kesalahan seperti halnya penggunaan variabel yang tidak terdefinisi

#### 5. Konstanta

Konstanta itu, seperti variabel biasa, namun... Dia mempunyai nilai tetap dan tidak dapat diubah nilainya setelah dideklarasikan. Seperti halnya dengan nilai phi, yang mana akan selalu tetap 22/7 atau 3,14

#### 6. Input

Input adalah kegiatan memasukkan data yang mana akan kita cari hasilnya nanti

#### 7. Output

Output adalah kegiatan yang mana setelah kita memasukkan nilai input beserta programnya seperti misalnya rumus kubus mungkin dengan rumus  $p.l.t$  dan setelah diinput variabelnya maka nantinya akan menghasilkan yang namanya Output

#### 8. Paradigma Perulangan

Perulangan merupakan salah satu struktur kontrol yang memungkinkan suatu instruksi yang sama dilakukan berulang kali dalam waktu atau jumlah yang lama. Tanpa instruksi perulangan, maka suatu instruksi akan ditulis dalam jumlah yang sangat banyak.

Sebagai contoh adalah menuliskan suatu teks "CAK1BAB3 Algoritma Pemrograman 1" sebanyak 1000 baris.

#### 9. Karakteristik For-Loop (Perulangan berdasarkan iterasi)

Salah satu instruksi perulangan yang paling mudah adalah **for-loop**, yang mana dengan instruksi ini dapat digunakan untuk mengulangi instruksi sebanyak ***n*** kali (iterasi). Batasan besar nilai dari ***n*** menyesuaikan dengan batasan dari tipe data integer yang digunakan.

Instruksi for-loop memiliki beberapa komponen, yaitu:

**1.inisialisasi** merupakan assignment **variabel iterasi** yang bertipe integer. Pada contoh di atas biasanya **variabel iterasi = 0** atau **1**, artinya iterasi dimulai dari 0 atau 1.

**2.kondisi** merupakan suatu operasi bernilai boolean yang menyatakan kapan perulangan harus dilakukan. Pada contoh di atas **kondisi** adalah **variabel iterasi <= n** (kurang dari atau sama dengan)

**3.update** merupakan ekspresi yang menyatakan perubahan nilai dari **variabel iterasi**.

Pada contoh di atas update adalah **variabel iterasi = variabel iterasi + 1**.

## REPEAT UNTIL

### Dasar Teori Repeat Until dalam Pemrograman Golang

"Repeat Until" adalah sebuah struktur kontrol dalam pemrograman yang memungkinkan kode untuk diulang berkali-kali sampai suatu kondisi tertentu terpenuhi.

#### Cara Kerjanya:

1. Kode di dalam blok "Repeat Until" dijalankan sekali.
2. Kondisi yang ditentukan diuji.
3. Jika kondisi benar, perulangan berhenti, dan program melanjutkan ke baris kode berikutnya setelah blok "Repeat Until".
4. Jika kondisi salah, program kembali ke awal blok "Repeat Until" dan mengulang kode di dalamnya.

#### Contoh dalam Pseudocode:

```
Repeat Until (kondisi == benar)
// Kode yang akan diulang
End Repeat
```

#### Keuntungan Repeat Until:

- Jaminan Eksekusi Minimal: Kode dalam blok "Repeat Until" setidaknya akan dijalankan sekali, bahkan jika kondisi sudah benar sejak awal.
- Kemudahan dalam Mengulang Kode: Struktur ini sederhana dan mudah digunakan untuk mengulang kode sampai kondisi tertentu terpenuhi.

#### Contoh Penggunaan:

Misal, Anda ingin meminta input dari pengguna sampai mereka memasukkan angka yang valid (misalnya, angka antara 1 dan 10). Anda dapat menggunakan "Repeat Until" seperti ini:

```
Repeat Until (angka >= 1 and angka <= 10)
Tampilkan "Masukkan angka antara 1 dan 10: "
Baca angka dari input pengguna
End Repeat
```

Dalam contoh ini, kode di dalam blok "Repeat Until" akan terus diulang sampai pengguna memasukkan angka yang berada dalam rentang 1 hingga 10.

#### Catatan:

- Pernyataan "Repeat Until" tidak tersedia di semua bahasa pemrograman. Bahasa seperti Golang menggunakan struktur "for" yang fleksibel untuk mensimulasikan fungsinya.
- Struktur yang mirip dengan "Repeat Until" adalah "Do While" yang akan mengecek kondisi setelah kode dijalankan sekali.

**Repeat until** secara langsung seperti dalam bahasa pemrograman lain seperti Pascal. Golang menggunakan pernyataan **for** yang fleksibel untuk menangani berbagai jenis perulangan, termasuk perulangan yang mirip dengan **repeat until**.

### **Perulangan for dalam Golang**

Pernyataan **for** dalam Golang adalah konstruksi yang kuat dan serbaguna untuk mengulang blok kode. Berikut adalah bentuk umum pernyataan **for** :

```
for initialization; condition; post-statement {  
// Blok kode yang akan diulang  
}
```

### **Meniru repeat until dengan for**

Untuk meniru perilaku **repeat until** dalam Golang, Anda dapat menggunakan pernyataan **for** dengan kondisi yang selalu benar ( **true** ) dan pernyataan **break** untuk keluar dari perulangan ketika kondisi tertentu terpenuhi. Berikut contohnya:

```
package main  
import "fmt"  
func main() {  
    i := 0  
    for true {  
        fmt.Println(i)  
        i++  
        if i > 5 {  
            break // Keluar dari perulangan ketika i lebih besar dari 5  
        }  
    }  
}
```

### **Penjelasan:**

1. **for true** : Perulangan ini akan terus berjalan karena kondisi **true** selalu benar.
2. **fmt.Println(i)** : Mencetak nilai **i** ke konsol.
3. **i++** : Meningkatkan nilai **i** dengan 1.
4. **if i > 5 { break }** : Jika nilai **i** lebih besar dari 5, pernyataan **break** akan dieksekusi, yang akan keluar dari perulangan **for**.

### **Kesimpulan**

Meskipun Golang tidak memiliki pernyataan **repeat until** secara langsung, Anda dapat meniru fungsinya menggunakan pernyataan **for** dengan kondisi **true** dan pernyataan **break**. Ini memungkinkan Anda untuk mengulang blok kode sampai kondisi tertentu terpenuhi, mirip dengan perilaku **repeat until**.

## CONTOH SOAL

Contoh 1:

Coding Program Contoh Soal 1:

```
contohsoal1 > go contohsoal1.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     var word string
7     var repetitions int
8     fmt.Scan(&word, &repetitions)
9     counter := 0
10    for done := false; !done; {
11        fmt.Println(word)
12        counter++
13        done = (counter >= repetitions)
14    }
15 }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\contohsoal1\contohsoal1.go"
pagi 3
pagi
pagi
pagi
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\contohsoal1\contohsoal1.go"
kursi 5
kursi
kursi
kursi
kursi
kursi
PS C:\Ryan\Tugas Alprog\praktikum9> 
```

Deskripsi program dari contoh soal 1:

Tujuan program contoh soal 1 adalah membuat program menggunakan bahasa Go yang menerima input kata dan mencetaknya sebanyak jumlah pengulangan yang diinginkan oleh pengguna. Program akan dihentikan ketika jumlah kata yang dicetak mencapai jumlah yang diinginkan oleh pengguna.

Contoh:

1.Input: pagi 3

Output: pagi pagi pagi

2.Input: pagi 5

Output: pagi pagi pagi pagi pagi

Contoh 2:

Coding Program Contoh Soal 2:

```
contohsoal2 > go contohsoal2.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var number int
7      var continueLoop bool
8      for continueLoop = true; continueLoop; {
9          fmt.Scan(&number)
10         continueLoop = number <= 0
11     }
12     fmt.Printf("%d adalah bilangan bulat positif\n", number)
13 }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\contohsoal2\contohsoal2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\contohsoal2\contohsoal2.go"
17
17 adalah bilangan bulat positif
PS C:\Ryan\Tugas Alprog\praktikum9> |
```

Deskripsi program dari contoh soal 2:

Tujuan program contoh soal 2 adalah membuat program dalam bahasa Go yang meminta pengguna untuk memasukkan bilangan bulat positif. Program akan terus meminta input hingga pengguna memasukkan bilangan bulat positif.

Contoh:

1.Input:

-5

-2

-1

0

5

Output: 5 adalah bilangan bulat positif

2.Input: 17

Output: 17 adalah bilangan bulat positif

Contoh 3:

Coding Program Contoh Soal :

```
contohsoal3 > go contohsoal3.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var x int
7      var y int
8      var selesai bool
9      fmt.Scan(&x, &y)
10     for selesai = false; !selesai; {
11         x = x - y
12         fmt.Println(x)
13         selesai = x <= 0
14     }
15     fmt.Println(x == 0)
16 }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\contohsoal3\contohsoal3.go"
5
2
3
1
-1
false
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\contohsoal3\contohsoal3.go"
15
3
12
9
6
3
0
true
PS C:\Ryan\Tugas Alprog\praktikum9>
```

Deskripsi program dari contoh soal 3:

Tujuan program contoh soal 3 adalah membuat program yang digunakan untuk melakukan pengecekan apakah suatu bilangan merupakan kelipatan dari bilangan lainnya.

Contoh:

1.Input: 5 2, Output: 3 1-1 false

2.Input: 15 3, Output: 12 9 6 3 0 true



## LATIHAN SOAL

Latihan Soal 1:

Coding Program Latihan Soal 1:

```
latihansoal1 > go latihansoal1.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var bilangan int
7      fmt.Scan(&bilangan)
8      jumlahDigit := 0
9
10     for {
11         jumlahDigit++
12         bilangan /= 10
13         if bilangan == 0 {
14             break
15         }
16     }
```

Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal1\latihansoal1.go"
5
1
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal1\latihansoal1.go"
234
3
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal1\latihansoal1.go"
78787
5
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal1\latihansoal1.go"
1894256
7
PS C:\Ryan\Tugas Alprog\praktikum9> |
```

Deskripsi program dari Latihan soal 1:

Tujuan program Latihan soal 1 adalah membuat program yang digunakan untuk menghitung banyaknya digit dari suatu bilangan.

Contoh:

1.Input: 5

Output: 1

2.Input: 234

Output: 3

3.Input: 78787

Output: 5

4.Input: 1894256

Output: 7

## Latihan Soal 2:

### Coding Program Latihan Soal 2:

```
latihansoal2 > go latihansoal2.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var x float64
7      var itg int
8      fmt.Scan(&x)
9      itg = int(x) + 1
10     for done := false; !done; {
11         x = x + 0.1
12         fmt.Printf("%.1f\n", x)
13         done = x > float64(itg)-0.11
14     }
15     fmt.Println(itg)
16 }
```

### Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal2\latihansoal2.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal2\latihansoal2.go"
2.7
2.8
2.9
3
PS C:\Ryan\Tugas Alprog\praktikum9> 
```

### Deskripsi program dari Latihan soal 2:

Tujuan program Latihan soal 2 adalah membuat suatu program yang digunakan untuk mendapatkan bilangan bulat optimal dari bilangan yang telah diinputkan. Melakukan penjumlahan tiap perulangan mencapai pembulatan ke atas dari bilangan yang diinputkan.

.Contoh:

1.Input: 0,2

Output: 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1

1.Input: 2,7

Output: 2,8; 2,9; 3

### Latihan Soal 3:

### Coding Program Latihan Soal 3:

```
latihansoal3 > go latihansoal3.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var target, donasi, totalDonasi, jumlahDonatur int
7      fmt.Scan(&target)
8
9      for totalDonasi < target {
10         fmt.Scan(&donasi)
11         jumlahDonatur++
12         totalDonasi += donasi
13         fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", jumlahDonatur, donasi, totalDonasi)
14     }
15     fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", totalDonasi, jumlahDonatur)
16 }
```

### Hasil Coding:

```
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal3\latihansoal3.go"
300
100
Donatur 1: Menyumbang 100. Total terkumpul: 100
50
Donatur 2: Menyumbang 50. Total terkumpul: 150
200
Donatur 3: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal3\latihansoal3.go"
500
150
Donatur 1: Menyumbang 150. Total terkumpul: 150
100
Donatur 2: Menyumbang 100. Total terkumpul: 250
50
Donatur 3: Menyumbang 50. Total terkumpul: 300
300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS C:\Ryan\Tugas Alprog\praktikum9> go run "c:\Ryan\Tugas Alprog\praktikum9\latihansoal3\latihansoal3.go"
200
300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
PS C:\Ryan\Tugas Alprog\praktikum9> █
```

### Deskripsi program dari Latihan soal 3:

Tujuan program Latihan soal 3 adalah membuat suatu program. Program ini akan terus meminta input dari pengguna untuk jumlah donasi hingga total donasi mencapai atau melebihi target yang telah ditentukan.

#### Contoh:

##### 1. Input:

300

100

50

200

Output:

Donatur 1: Menyumbang 100. Total terkumpul: 100

Donatur 2: Menyumbang 50. Total terkumpul: 150

Donatur 3: Menyumbang 200. Total terkumpul: 350

Target tercapai! Total donasi: 350 dari 3 donatur.

2.Input:

500

150

100

50

300

Output:

Donatur 1: Menyumbang 150. Total terkumpul: 150

Donatur 2: Menyumbang 100. Total terkumpul: 250

Donatur 3: Menyumbang 50. Total terkumpul: 300 Donatur 4: Menyumbang 300. Total terkumpul: 600

Target tercapai! Total donasi: 600 dari 4 donatur.

3.Input:

200

300

Output:

Donatur 1: Menyumbang 300. Total terkumpul: 300

Target tercapai! Total donasi: 300 dari 1 donatur.