

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 14
“KOMPOSISI”



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

Komposisi adalah konsep hebat dalam pemrograman berorientasi objek yang memungkinkan pembuatan tipe kompleks dengan menggabungkan tipe yang lebih kecil dan lebih sederhana. Komposisi memungkinkan pembuatan tipe baru yang memiliki fungsionalitas dari beberapa tipe yang sudah ada, tanpa perlu mewarisinya. Dalam Go, komposisi dicapai dengan menggunakan struct embedding, fitur bahasa yang memungkinkan penyisipan satu tipe struct di dalam tipe lain.

A. Apa itu Komposisi?

Komposisi adalah mekanisme yang memungkinkan penggabungan tipe yang lebih sederhana untuk menciptakan tipe yang lebih kompleks. Ini adalah konsep dasar dalam pemrograman berorientasi objek yang memungkinkan penggunaan kembali kode dan membangun program yang lebih fleksibel dan modular. Komposisi sering digunakan untuk memecahkan masalah yang timbul dari penggunaan pewarisan, seperti masalah kelas dasar yang rapuh atau masalah berlian. Tidak seperti pewarisan, komposisi didasarkan pada gagasan hubungan "memiliki", bukan hubungan "adalah". Itu berarti bahwa suatu tipe yang terdiri dari beberapa tipe lain memiliki fungsionalitas dari tipe-tipe tersebut, tetapi bukan merupakan subtype dari tipe-tipe tersebut.

Dalam Go, komposisi dicapai dengan menggunakan struct embedding, fitur bahasa yang memungkinkan penyisipan satu tipe struct di dalam tipe lain. Ini memungkinkan tipe yang dibuat untuk mewarisi kolom dan metode dari tipe yang disematkan, dan menggunakannya seolah-olah itu miliknya sendiri.

B. Komposisi dengan Penanaman Struktur

Penyematan struktur merupakan mekanisme utama untuk mencapai komposisi dalam Go. Mekanisme ini memungkinkan penyematan satu tipe struktur di dalam tipe lain, sehingga menciptakan tipe baru yang memiliki kolom dan metode dari kedua tipe tersebut. Tipe yang disematkan disebut sebagai struktur yang disematkan, dan tipe penyematan disebut sebagai struktur penyematan.

C. Komposisi vs. Pewarisan

Komposisi dan pewarisan adalah dua cara untuk mencapai penggunaan kembali kode dan membangun sistem yang kompleks. Pewarisan adalah mekanisme yang memungkinkan kelas mewarisi properti dan perilaku dari kelas induk. Komposisi adalah cara membangun objek kompleks dengan menggabungkan objek yang lebih kecil dan lebih sederhana. Dalam Go,

komposisi lebih disukai daripada pewarisan. Hal ini karena Go tidak memiliki kelas seperti bahasa pemrograman berorientasi objek tradisional. Sebagai gantinya, Go menggunakan struktur untuk mendefinisikan objek dan antarmuka untuk mendefinisikan perilaku. Komposisi dicapai dengan menanamkan satu struktur ke struktur lainnya.

CONTOH SOAL

1. Latihan1

Buatlah program yang digunakan untuk menampilkan sejumlah bilangan ganjil.

Masukan terdiri dari sebuah bilangan bulat. Keluaran berupa barisan bilangan ganjil dari 1 hingga bilangan bulat yang diberikan pada masukan.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1.	3	1 3
2.	2	1
3.	7	1 3 5 7
4.	10	1 3 5 7 9

Source Code:

```
package main

import "fmt"

func main() {
    var bilangan, j int
    fmt.Scan(&bilangan)
    for j = 1; j<=bilangan; j+=1{
        if j%2 != 0 {
            fmt.Print(j, " ")
        }
    }
}
```

Output:

```
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./coso1.go"
3
1 3 %
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./coso1.go"
2
1 %
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./coso1.go"
7
1 3 5 7 %
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./coso1.go"
10
1 3 5 7 9 %
○ mymac@192 praktikum1 %
```

Deskripsi Program:

- Var bilangan dan j mendeklarasikan variabel dengan tipe data integer
- Fmt.Scan(&bilangan) untuk menginputkan isi dari variabel bilangan
- For j = 1| j <= bilangan; j+=1 {...} melakukan perulangan Ketika nilai j mulai dari 1 dan kurang dari bilangan
- if j%2 != 0: Mengecek apakah j adalah bilangan ganjil (dengan modulus %2).
- Jika true (j adalah bilangan ganjil), program mencetak j dengan fungsi fmt.Print.

2. Latihan 2

Buatlah program Go yang digunakan untuk mencari nilai terbesar dan terkecil antara tiga bilangan yang diberikan. Masukan terdiri dari 3 bilangan bulat. Keluaran terdiri dari dua bilangan yang menyatakan bilangan terbesar dan terkecil dari tiga bilangan yang diberikan.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1.	1 2 3	Terbesar 3 Terkecil 1
2.	5 5 5	Terbesar 5 Terkecil 5
3.	12 31 -43	Terbesar 31 Terkecil -43

Source Code :

```
package main

import "fmt"

func main() {
    var b1,b2,b3,max,min int
    fmt.Scan(&b1,&b2,&b3)
    if b1 > b2 {
        max = b1
        min = b2
    }else{
        max = b2
        min = b1
    }
}
```

```

}
if max < b2{
    max = b3
}
if min > b3{
    min = b3
}
fmt.Println("Terbesar", max)
fmt.Println("Terkecil", min)
}

```

Output:

```

● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./tempCodeRunnerFile.go"
1 2 3
Terbesar 2
Terkecil 1
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./tempCodeRunnerFile.go"
5 5 5
Terbesar 5
Terkecil 5
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./tempCodeRunnerFile.go"
12 31 -43
Terbesar 31
Terkecil -43
○ mymac@192 praktikum1 %

```

Deskripsi Program:

- Var b1,b2,b3,max, min mendeklarasikan variabel dengan tipe data integer
- Fmt.Scan(&b1,&b2,&b3) untuk menginputkan nilai masing-masing variabel
- if b1 > b2 { max = b1; min = b2 } else { max = b2; min = b1 }: Membandingkan b1 dan b2, untuk menentukan max dan min.
- Jika b1 lebih besar dari b2, maka max diisi dengan b1 dan min diisi dengan b2. Jika tidak, max diisi dengan b2 dan min diisi dengan b1.
- if max < b3 { max = b3 }: Membandingkan nilai max dengan b3. Jika b3 lebih besar dari max, maka max diubah menjadi b3.
- if min > b3 { min = b3 }: Membandingkan nilai min dengan b3. Jika b3 lebih kecil dari min, maka min diubah menjadi b3.
- fmt.Println("Terbesar", max): Menampilkan nilai maksimum.

- `fmt.Println("Terkecil", min)`: Menampilkan nilai minimum.

3. Latihan 3

Buatlah program yang digunakan untuk menampilkan faktor bilangan dari suatu nilai. Faktor bilangan dari x adalah himpunan bilangan yang habis membagi bilangan x. Masukan terdiri dari suatu bilangan bulat positif. Keluaran berupa baris bilangan yang menyatakan semua faktor bilangan dari bilangan yang diberikan pada masukan.**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1.	5	1 5
2.	12	1 2 3 4 6 12
3.	20	1 2 4 5 10 20
4.	72	1 2 3 4 6 8 9 12 18 24 36 72

Source Code

```
package main

import "fmt"

func main() {
    var bilangan, j int
    fmt.Scan(&bilangan)
    for j = 1; j<=bilangan; j+=1{
        if bilangan%j == 0 {
            fmt.Print(j, " ")
        }
    }
}
```

Output:

```

● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./tempCodeRunnerFile.go"
5
1 5 %
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./tempCodeRunnerFile.go"
12
1 2 3 4 6 12 %
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./tempCodeRunnerFile.go"
20
1 2 4 5 10 20 %
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./tempCodeRunnerFile.go"
72
1 2 3 4 6 8 9 12 18 24 36 72 %
○ mymac@192 praktikum1 %

```

Deskripsi Program:

- Var bilangan dan j mendeklarasikan 2 buah variabel dengan tipe data integer
- Fmt.Scan(&bilangan) digunakan untuk menginputkan nilai dari var bilangan
- For j = 1 | j <= bilangan; j+=1 {...} melakukan perulangan Ketika nilai j mulai dari 1 dan kurang dari bilangan
- If bilangan % j == 0 (jika bilangan habis dibagi j) Jika true, artinya j adalah pembagi bilangan, maka program mencetak j.

SOAL LATIHAN

Statement Percabangan

1. Latihan Soal Nomor 2

Sebuah program digunakan untuk menentukan sebuah bilangan adalah prima atau bukan. Bilangan dikatakan prima apabila hanya memiliki faktor yaitu satu dan bilangan itu sendiri. Sebagai catatan bilangan satu bukanlah bilangan prima.

Masukan terdiri dari suatu bilangan bulat positif.

Keluaran berupa teks yang menyatakan bilangan adalah "prima" atau "bukan prima".

Contoh masukan dan keluaran

No	Masukan	Keluaran
1.	5	Prima
2.	12	Bukan prima
3.	19	Prima
4.	72	Bukan prima

Source Code:

```
package main

import "fmt"

func main() {
    var n int
    var result string

    fmt.Scan(&n)
    result = "prima"

    if n%n == 0 && n%2 != 0 {
        fmt.Println(result)
    }else{
        fmt.Println("Bukan Prima")
    }
}
```

Output:

```

● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol1.go"
5
prima
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol1.go"
12
Bukan Prima
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol1.go"
19
prima
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol1.go"
72
Bukan Prima

```

Deskripsi Program:

- var n mendeklarasikan variabel dengan tipe data integer
- var result mendeklarasikan variabel dengan tipe data string
- fmt.Scan(&n) untuk menginputkan nilai var n
- result = "prima" menginisiasikan result dengan nilai prima
- if n%n == 0 %% n%2 != 0 melakukan percabangan, Ketika nilai n habis dibagi dirinya sendiri dan n tidak habis dibagi 2 maka akan melakukan print result.
- Jika kondisi tidak terpenuhi maka akan melakukan print ("Bukan prima)

2. Latihan Soal nomor 1

Buatlah sebuah program Go yang digunakan untuk menghitung banyaknya bilangan ganjil dari 1 hingga n.

Masukan terdiri dari suatu bilangan bulat positif n.

Keluaran terdiri dari suatu teks yang menyatakan banyaknya bilangan ganjil yang terdapat antara 1 hingga n.

Catatan: Gunakan perulangan untuk pengecekan bilangan, bukan menggunakan operasi aritmatika. (adalah input/read):

Contoh masukan dan keluaran

No	Masukan	Keluaran
1.	3	Terdapat 2 bilangan ganjil
2.	2	Terdapat 1 bilangan ganjil
3.	7	Terdapat 4 bilangan ganjil
4.	10	Terdapat 5 bilangan ganjil

Source Code:

```

package main

import "fmt"

func main() {
    var n, i, jumlahGanjil int

    fmt.Scan(&n)

    for i = 1; i <= n; i++ {
        if i%2 != 0 {
            jumlahGanjil++
        }
    }

    fmt.Printf("Terdapat %d bilangan ganjil\n", jumlahGanjil)
}

```

Output:

```

• mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol2.go"
3
Terdapat 2 bilangan ganjil
• mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol2.go"
2
Terdapat 1 bilangan ganjil
• mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol2.go"
7
Terdapat 4 bilangan ganjil
• mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul14./latsol2.go"
10
Terdapat 5 bilangan ganjil

```

Deskripsi Program :

- var n, i, jumlah ganjil mendeklarasikan variabel dengan tipe data integer
- fmt.Scan(&n) untuk menginputkan nilai n
- for i = 1; i <= n; i++ melakukan perulangan jika nilai i tidak habis dibagi 2 maka jumlahganjil ditambah 1

DAFTAR PUSTAKA

(N.d.). Retrieved from https://www-tutorialspoint-com.translate.goog/composition-in-golang?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc