

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 14
“KOMPOSISI”



DISUSUN OLEH:
ANGGUN WAHYU WIDIYANA
103112480280
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

Pengantar Komposisi

Komposisi merupakan salah satu konsep fundamental yang menjadi alternatif dari pewarisan (inheritance) yang umum ditemui pada bahasa pemrograman berorientasi objek lainnya. Go menggunakan pendekatan "composition over inheritance" yang berarti lebih mengutamakan komposisi dibanding pewarisan.

Konsep dasar komposisi

Komposisi adalah teknik pemrograman dimana sebuah struct dapat menyertakan struct lain sebagai bagian dari propertinya. Ini memungkinkan untuk membangun struktur data yang kompleks dengan menggabungkan beberapa struct yang lebih sederhana.

Kondisi Keuntungan Komposisi dalam Go:

1. Fleksibilitas yang lebih tinggi dibandingkan pewarisan karena memungkinkan penggabungan berbagai komponen secara lebih dinamis
2. Menghindari masalah yang sering muncul dalam hierarki pewarisan yang kompleks
3. Memudahkan dalam pengujian dan pemeliharaan kode
4. Mendukung prinsip "has-a" relationship dibanding "is-a" relationship

Implementasi menggunakan Go

Komposisi diimplementasikan dengan cara menyertakan sebuah struct tanpa nama field di dalam struct lain. Contoh dasarnya seperti:

```
type Engine struct {  
    Power int  
    Type  string  
}  
  
type Car struct {  
    Engine // Komposisi: Car memiliki semua properti Engine  
    Brand  string  
    Model  string  
}
```

Interface dalam Komposisi:

Go juga mendukung komposisi interface, dimana sebuah interface dapat terdiri dari beberapa interface lain. Ini memungkinkan pembuatan interface yang lebih kompleks dari interface-interface yang lebih sederhana.

```
type Reader interface {
    Read(p []byte) (n int, err error)
}

type Writer interface {
    Write(p []byte) (n int, err error)
}

type ReadWriter interface {
    Reader
    Writer
}
```

Promotion dalam Komposisi:

Salah satu fitur penting dalam komposisi Go adalah promotion, dimana metode dan field dari embedded struct secara otomatis dipromosikan ke struct yang menyertainya. Ini memungkinkan akses langsung ke metode dan field tersebut tanpa perlu referensi eksplisit.

Prinsip-prinsip Penting:

1. Single Responsibility Principle: Setiap struct sebaiknya memiliki satu tanggung jawab spesifik
 2. Interface Segregation: Lebih baik memiliki banyak interface kecil daripada satu interface besar
 3. Loose Coupling: Mengurangi ketergantungan antar komponen
 4. High Cohesion: Memastikan komponen-komponen yang terkait berada bersama
-

CONTOH SOAL

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.

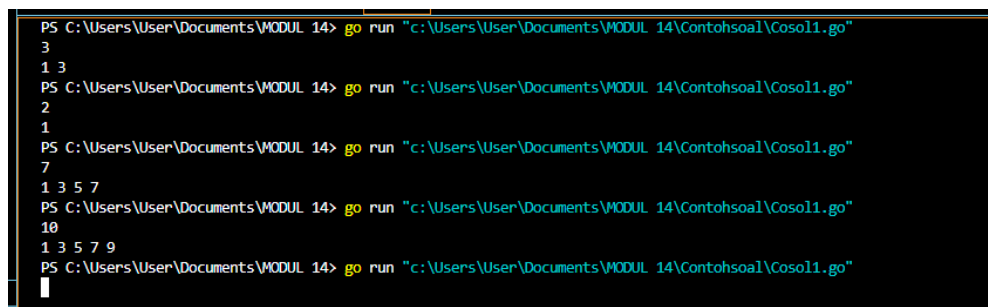
Contoh masukan dan keluaran:

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:



```
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Cosol1.go"
3
1 3
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Cosol1.go"
2
1
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Cosol1.go"
7
1 3 5 7
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Cosol1.go"
10
1 3 5 7 9
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Cosol1.go"
10
1 3 5 7 9
```

Deskripsi Program:

Program Go ini berfungsi untuk menampilkan bilangan ganjil dari 1 hingga bilangan bulat yang diberikan sebagai input oleh pengguna. Program meminta pengguna untuk memasukkan sebuah bilangan bulat, kemudian menggunakan loop untuk iterasi dari 1 hingga bilangan tersebut. Di dalam loop, program

memeriksa apakah angka saat ini adalah ganjil dengan menggunakan operator modulo. Jika angka tersebut ganjil, program mencetaknya ke layar dengan spasi sebagai pemisah. Hasilnya adalah deretan bilangan ganjil yang dimulai dari 1 hingga nilai yang ditentukan oleh pengguna.

Latihan2

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 Terkeci 1
2	5 5 5	Terbesar 5 Terkeci 5
3	12 31 -43	Terbesar 31 Terkeci -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 < b3 {
        max = b3
    }
    if min > b3 {
        min = b3
    }
    fmt.Println("Terbesar", max)
    fmt.Println("Terkecil", min)
}
```

Output:

```
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso12.go"
1 2 3
Terbesar 3
Terkecil 1
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso12.go"
5 5 5
Terbesar 5
Terkecil 5
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso12.go"
12 31 -43
Terbesar 31
Terkecil -43
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso12.go"
```

Deskripsi Program:

Program Go ini berfungsi untuk mencari nilai terbesar dan terkecil dari tiga bilangan bulat yang diberikan oleh pengguna. Program meminta pengguna untuk memasukkan tiga bilangan bulat, kemudian membandingkan nilai-nilai tersebut untuk menentukan mana yang merupakan bilangan terbesar dan terkecil. Setelah melakukan perbandingan, program mencetak hasilnya, menampilkan bilangan terbesar dan terkecil ke layar. Hasil keluaran memberikan informasi yang jelas tentang nilai maksimum dan minimum dari tiga bilangan yang dimasukkan.

Latihan3

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:

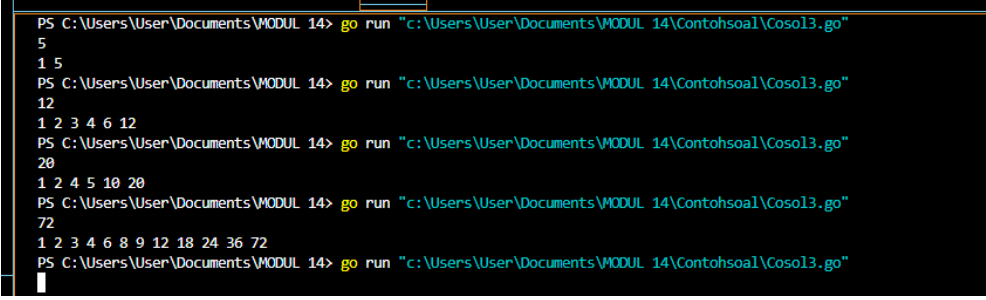
```
package main

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:



```
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso13.go"  
5  
1 5  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso13.go"  
12  
1 2 3 4 6 12  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso13.go"  
28  
1 2 4 5 10 20  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso13.go"  
72  
1 2 3 4 6 8 9 12 18 24 36 72  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Contohsoal\Coso13.go"
```

Deskripsi Program:

Program Go ini berfungsi untuk menampilkan semua faktor dari suatu bilangan bulat positif yang diberikan oleh pengguna. Program meminta pengguna untuk memasukkan bilangan bulat positif dan kemudian menggunakan loop untuk iterasi dari 1 hingga bilangan tersebut. Di dalam loop, program memeriksa apakah angka saat ini adalah faktor dengan menggunakan operator modulo. Jika angka tersebut adalah faktor, program mencetaknya ke layar dengan spasi sebagai pemisah. Hasil keluaran adalah deretan bilangan yang merupakan faktor dari bilangan yang dimasukkan oleh pengguna.

SOAL LATIHAN

Latihan1

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.

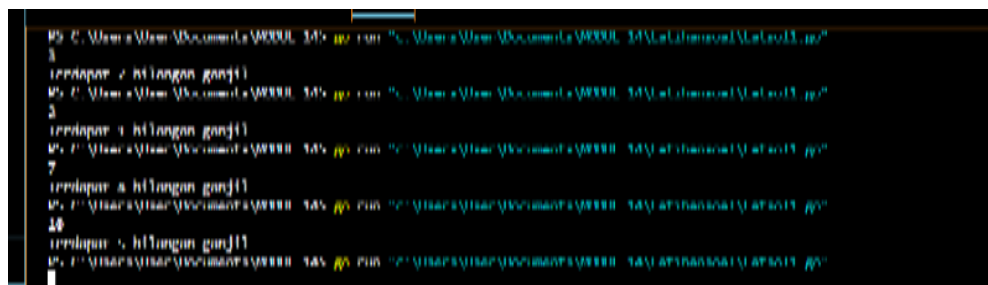
Contoh masukan dan keluaran

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 bilangan int
    fmt.Scan(&bilangan)
    if bilangan <= 0 {
        fmt.Println("bilangan bulat harus positif.")
        return
    }
    jumlahGanjil := 0
    for i := 1; i <= bilangan; i++ {
        if i%2 != 0 {
            jumlahGanjil++
        }
    }
    fmt.Printf("Terdapat %d bilangan ganjil\n", jumlahGanjil)
}
```

Output:



```
PS C:\Users\Wan\Documents\GOOL ID> go run "C:\Users\Wan\Documents\GOOL ID\Latihan1\Lat1.go"
3
Terdapat 2 bilangan ganjil
PS C:\Users\Wan\Documents\GOOL ID> go run "C:\Users\Wan\Documents\GOOL ID\Latihan1\Lat1.go"
2
Terdapat 1 bilangan ganjil
PS C:\Users\Wan\Documents\GOOL ID> go run "C:\Users\Wan\Documents\GOOL ID\Latihan1\Lat1.go"
7
Terdapat 4 bilangan ganjil
PS C:\Users\Wan\Documents\GOOL ID> go run "C:\Users\Wan\Documents\GOOL ID\Latihan1\Lat1.go"
10
Terdapat 5 bilangan ganjil
PS C:\Users\Wan\Documents\GOOL ID> go run "C:\Users\Wan\Documents\GOOL ID\Latihan1\Lat1.go"
```


Deskripsi Program:

Program Go ini untuk menghitung jumlah bilangan ganjil dari 1 hingga bilangan bulat positif yang dimasukkan oleh pengguna.

Setelah pengguna memasukkan nilai, program memeriksa apakah nilai tersebut adalah bilangan bulat positif. Jika tidak, program akan menampilkan pesan bahwa bilangan harus positif dan berhenti. Jika nilai valid, program akan melakukan iterasi dari 1 hingga nilai yang dimasukkan, menghitung berapa banyak bilangan ganjil yang ada dalam rentang tersebut. Akhirnya, program mencetak jumlah total bilangan ganjil yang ditemukan.

Latihan2

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 bilangan int
    fmt.Scan(&bilangan)

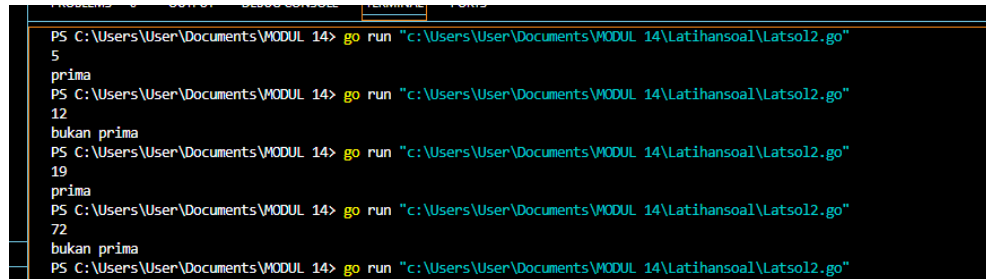
    if bilangan < 2 {
        fmt.Printf("%d bukan prima\n", bilangan)
        return
    }

    statusPrima := true
    for i := 2; i*i <= bilangan; i++ {
        if bilangan%i == 0 {
            statusPrima = false
            break
        }
    }

    if statusPrima {
        fmt.Printf("prima")
    }
}
```

```
    } else {  
        fmt.Printf("bukan prima")  
    }  
}
```

Output:



```
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latsol2.go"  
5  
prima  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latsol2.go"  
12  
bukan prima  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latsol2.go"  
19  
prima  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latsol2.go"  
72  
bukan prima  
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latsol2.go"
```

Deskripsi Program:

Program Go digunakan untuk menentukan apakah sebuah bilangan bulat positif adalah bilangan prima.

Setelah pengguna memasukkan nilai, program memeriksa apakah nilai tersebut kurang dari 2. Jika iya, program menyatakan bahwa bilangan tersebut "bukan prima". Jika bilangan valid (lebih besar atau sama dengan 2), program mengasumsikan bahwa bilangan tersebut adalah prima dan kemudian memeriksa faktor-faktor dari 2 hingga akar kuadrat dari bilangan tersebut. Jika ditemukan faktor yang membagi bilangan tersebut habis, status prima diubah menjadi "false".

Akhirnya, program mencetak hasilnya, menyatakan apakah bilangan tersebut "prima" atau "bukan prima" berdasarkan hasil pemeriksaan.

Latihan3

Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berturut-turut adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang.

Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan **true** apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan **false** untuk urutan warna lainnya. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

```
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true

Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
```

Source Code:

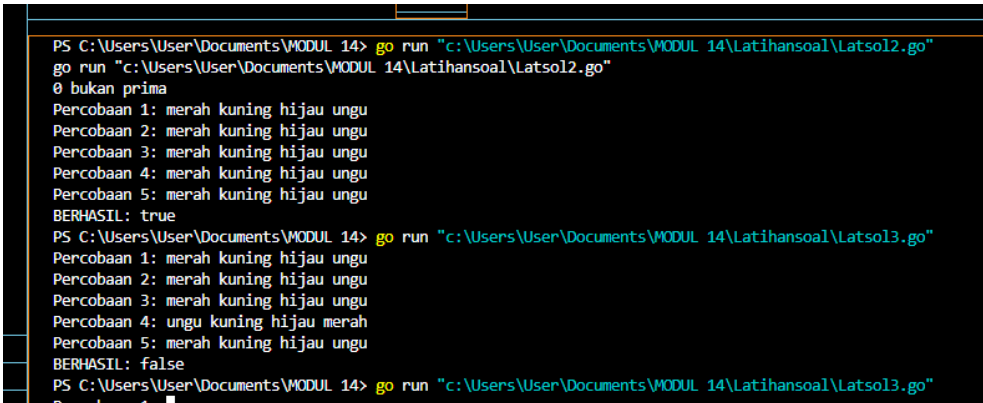
```
package main
import "fmt"
func main(){
    var warna1, warna2, warna3, warna4 string
    berhasil := true

    for percobaan := 1; percobaan <= 5; percobaan++ {
        fmt.Printf("Percobaan %d: ", percobaan)
        fmt.Scan(&warna1, &warna2, &warna3, &warna4)

        if warna1 != "merah" || warna2 != "kuning" || warna3
        != "hijau" || warna4 != "ungu" {
            berhasil = false
        }
    }

    fmt.Printf("BERHASIL: %t\n", berhasil)
}
```

Output:



```
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latso12.go"
go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latso12.go"
0 bukan prima
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latso13.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
PS C:\Users\User\Documents\MODUL 14> go run "c:\Users\User\Documents\MODUL 14\Latihansoal\Latso13.go"
Percobaan 1:
```

Deskripsi Program:

Program Go ini untuk untuk menentukan apakah hasil percobaan kimia berhasil berdasarkan urutan warna cairan di empat tabung reaksi.

Program meminta pengguna untuk memasukkan warna dari keempat tabung reaksi selama lima percobaan. Warna yang diharapkan dalam urutan yang benar adalah 'merah', 'kuning', 'hijau', dan 'ungu'. Jika semua percobaan mengikuti urutan ini, program akan menyatakan bahwa percobaan tersebut berhasil dengan menampilkan true. Namun, jika ada satu percobaan yang tidak sesuai dengan urutan yang ditentukan, program akan menampilkan false.

Hasil akhir ditampilkan setelah semua percobaan selesai, memberikan informasi tentang keberhasilan percobaan secara keseluruhan.

DAFTAR PUSTAKA

School of Computing. *Modul Praktikum 14 – Komposisi. Algoritma dan Pemrograman 1 SI Informatika*.2024