**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN 1**

**MODUL 14**

**“KOMPOSISI”**



**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 maksut wadah itu? Maksutnya 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 makan 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**.

**KOMPOSISI**

**Dasar Teori Komposisi dalam Pemrograman Golang  
Komposisi dalam Go (Golang) adalah teknik pemrograman yang memungkinkan Anda membuat struktur data yang kompleks dengan menggabungkan struktur data yang lebih sederhana. Ini mirip dengan konsep "has-a" dalam pemrograman berorientasi objek.  
  
Dasar Teori Komposisi dalam Golang  
- Struktur Data Embedded: Komposisi dalam Go dicapai dengan menanamkan (embedding) struktur data lain dalam struktur data Anda. Ini berarti bahwa struktur data yang ditanamkan menjadi bagian dari struktur data yang menanamkannya.  
- Akses Field: Anda dapat mengakses field dari struktur data yang ditanamkan secara langsung menggunakan nama field.  
  
Contoh Sederhana  
package main  
  
import "fmt"  
  
type Shape interface {  
Area() float64  
}  
  
type Rectangle struct {  
Width float64  
Height float64  
}  
  
func (r Rectangle) Area() float64 {  
return r.Width \* r.Height  
}  
  
type Circle struct {  
Radius float64  
}  
  
func (c Circle) Area() float64 {  
return 3.14159 \* c.Radius \* c.Radius  
}  
  
type ColoredShape struct {  
Shape Shape  
Color string  
}  
  
func main() {  
rect := Rectangle{Width: 5, Height: 10}  
circle := Circle{Radius: 5}  
redRect := ColoredShape{Shape: rect, Color: "Red"}  
blueCircle := ColoredShape{Shape: circle, Color: "Blue"}  
fmt.Println("Red Rectangle Area:", redRect.Shape.Area()) // Akses field Shape  
fmt.Println("Blue Circle Area:", blueCircle.Shape.Area()) // Akses field Shape  
}  
   
  
Penjelasan  
1. Struktur Data  Shape : Interface  Shape  mendefinisikan metode  Area()  yang harus diimplementasikan oleh setiap struktur data yang ingin dianggap sebagai "Shape".  
2. Struktur Data  Rectangle  dan  Circle : Kedua struktur data ini mengimplementasikan interface  Shape  dengan metode  Area()  yang sesuai.  
3. Struktur Data  ColoredShape : Struktur data ini menanamkan struktur data  Shape  menggunakan keyword  Shape Shape . Ini berarti bahwa  ColoredShape  "memiliki" sebuah  Shape .  
4. Akses Field: Dalam fungsi  main() , kita membuat objek  redRect  dan  blueCircle  dari tipe  ColoredShape . Kita kemudian dapat mengakses field  Shape  secara langsung menggunakan  redRect.Shape.Area()  dan  blueCircle.Shape.Area() .  
  
Keuntungan Komposisi  
- Reusability: Anda dapat menggunakan kembali struktur data yang ditanamkan di berbagai bagian kode Anda.  
- Modularitas: Kode Anda menjadi lebih modular karena Anda dapat memecah struktur data kompleks menjadi bagian-bagian yang lebih kecil.  
- Fleksibilitas: Anda dapat dengan mudah mengubah perilaku struktur data Anda dengan mengganti struktur data yang ditanamkan.**

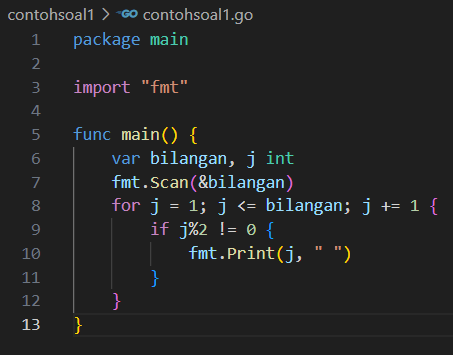
**Contoh Lain  
- Situs Web: Anda dapat menanamkan struktur data  User  dalam struktur data  Post  untuk mewakili penulis postingan.  
- Game: Anda dapat menanamkan struktur data  Position  dalam struktur data  Character  untuk mewakili lokasi karakter dalam game.**

**Kesimpulan  
Komposisi dalam Go adalah teknik yang kuat untuk membangun struktur data kompleks dengan memanfaatkan struktur data yang lebih sederhana. Ini meningkatkan modularitas, reusability, dan fleksibilitas kode Anda.**

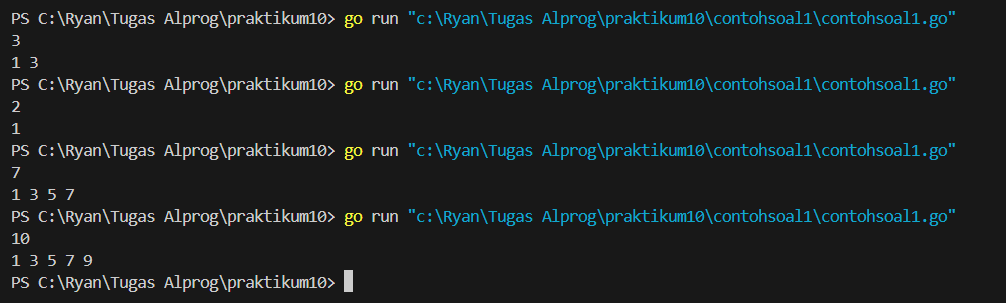
**CONTOH SOAL**

Contoh 1:

Coding Program Contoh Soal 1:



Hasil Coding:



Deskripsi program dari contoh soal 1:

Tujuan program contoh soal 1 adalah membuat program program yang digunakan untuk menampilkan sejumlah bilangan ganjil.

Contoh:

1.Input: 3

Output: 1 3

2.Input: 2

Output: 1

3.Input: 7

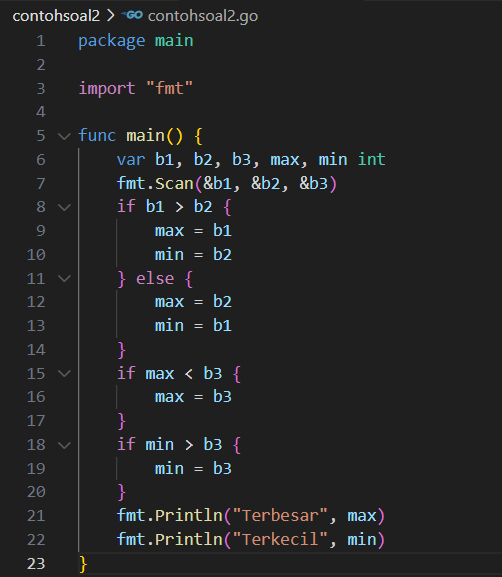
Output: 1 3 5 7

4.Input: 10

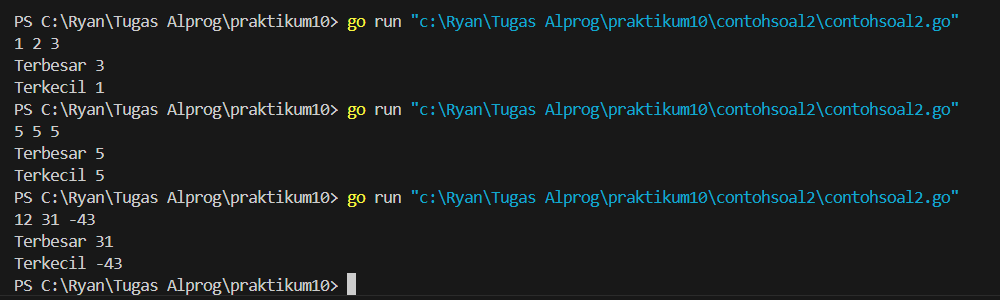
Output: 1 3 5 7 9

Contoh 2:

Coding Program Contoh Soal 2:



Hasil Coding:



Deskripsi program dari contoh soal 2:

Tujuan program contoh soal 2 adalah membuat program Go yang digunakan untuk mencari nilai terbesar dan terkecil antara tiga bilangan yang diberikan.

Contoh:

1.Input: 1 2 3

Output: Terbesar 3 Terkecil 1

2.Input: 5 5 5

Output: Terbesar 3 Terkecil 5

3.Input: 12 31 -43

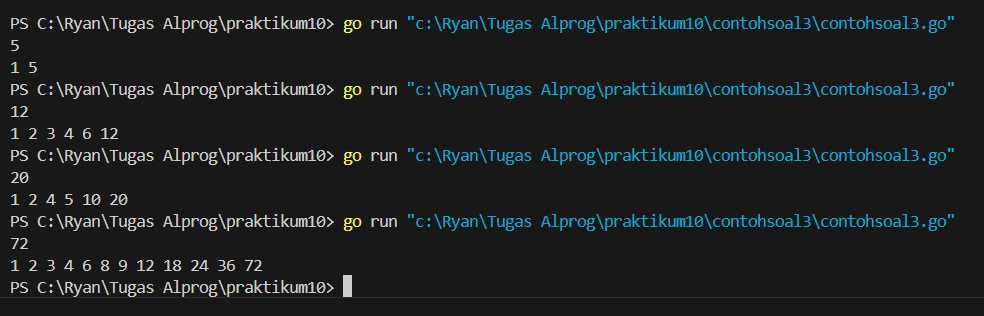
Output: Terbesar 31 Terkecil -43

Contoh 3:

Coding Program Contoh Soal :



Hasil Coding:



Deskripsi program dari contoh soal 3:

Tujuan program contoh soal 3 adalah membuat program yang digunakan untuk menampilkan faktor bilangan dari suatu nilai. Faktor bilangan dari x adalah himpunan bilangan yang habis membagi bilangan x.

Contoh:

1.Input: 5, Output: 1 5

2.Input: 12, Output: 1 2 3 4 6 12

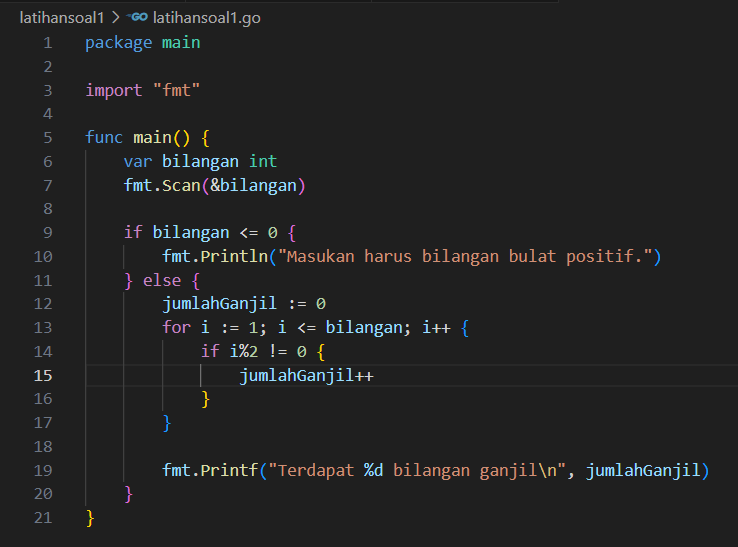
3.Input: 20, Output: 1 2 4 5 10 20

4.Input: 72, Output: 1 2 3 4 6 8 9 12 18 24 36 72

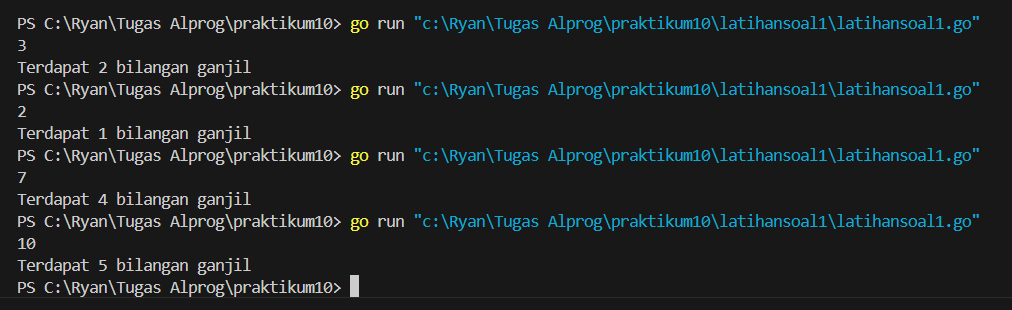
**LATIHAN SOAL**

Latihan Soal 1:

Coding Program Latihan Soal 1:



Hasil Coding:



Deskripsi program dari Latihan soal 1:

Tujuan program Latihan soal 1 adalah membuat program Go yang digunakan untuk menghitung banyaknya bilangan ganjil dari 1 hingga n.

Contoh:

1.Input: 3, Output: Terdapat 2 bilangan ganjil

2.Input: 2, Output: Terdapat 1 bilangan ganjil

3.Input: 3, Output: Terdapat 4 bilangan ganjil

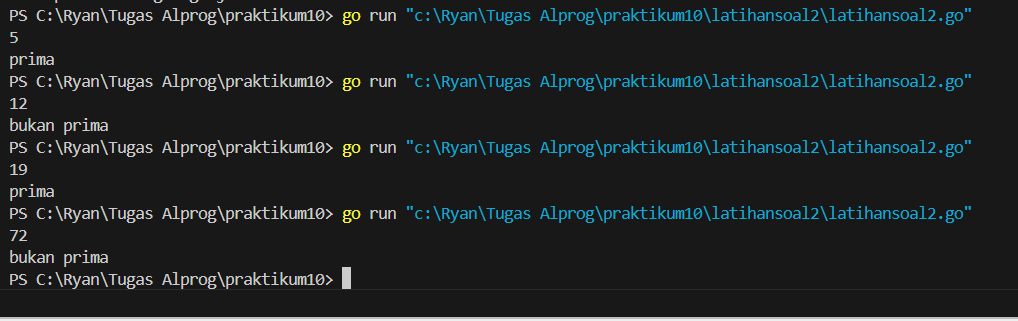
4.Input: 10, Output: Terdapat 5 bilangan ganjil

Latihan Soal 2:

Coding Program Latihan Soal 2:



Hasil Coding:



Deskripsi program dari Latihan soal 2:

Tujuan program Latihan soal 2 adalah membuat 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..

Contoh:

1.Input: 5, Output: prima

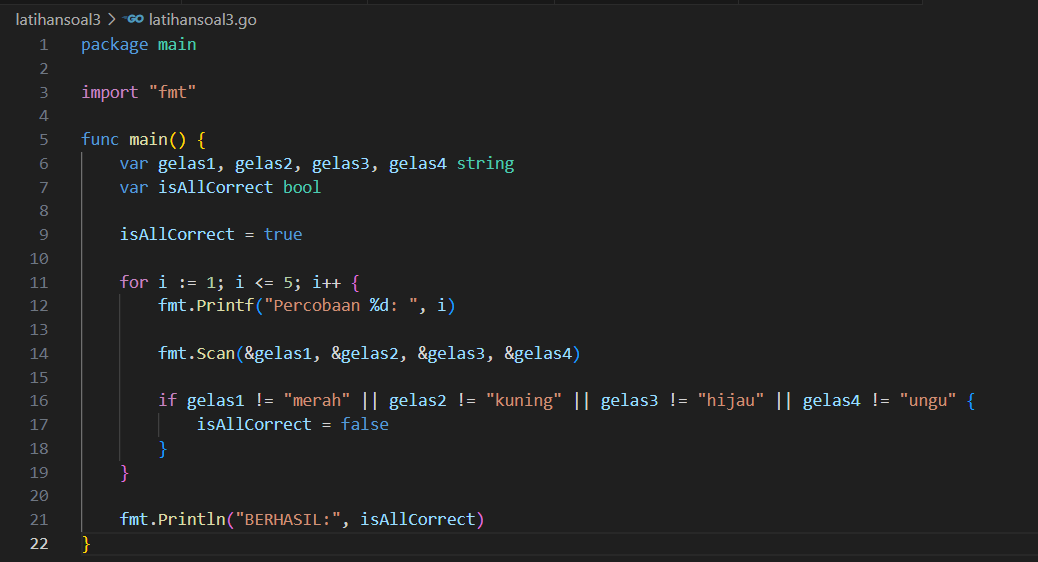
2.Input: 12, Output: bukan prima

3.Input: 19, Output: prima

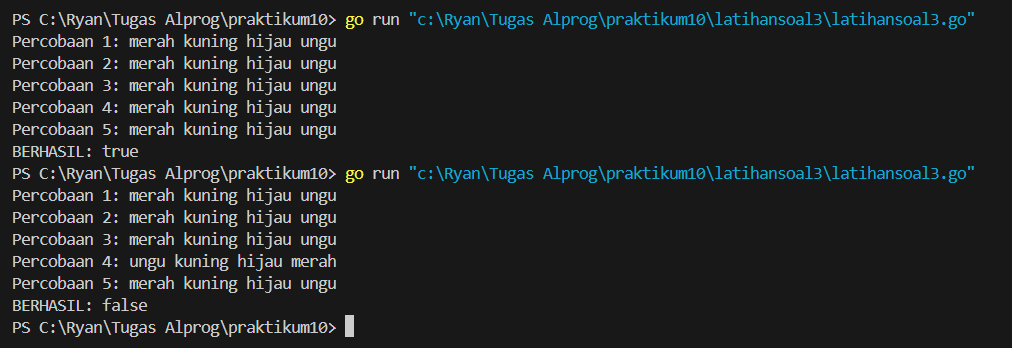
4.Input: 72, Output: bukan prima

Latihan Soal 3:

Coding Program Latihan Soal 3:



Hasil Coding:



Deskripsi program dari Latihan soal 3:

Tujuan program Latihan soal 3 adalah membuat 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.

Contoh:

1.Input:

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

Output:

BERHASIL: true

2.Input:

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

Output:

BERHASIL: false