

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

RAFFI YULIANSYAH

109082500101

S1IF-13-07

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

SOAL

1. SOAL 1

Source Code

```
package main

import "fmt"

func main(){

    var n, ganjil int

    fmt.Scan(&n)

    ganjil = 1

    for i := 1; i <= n; i++){

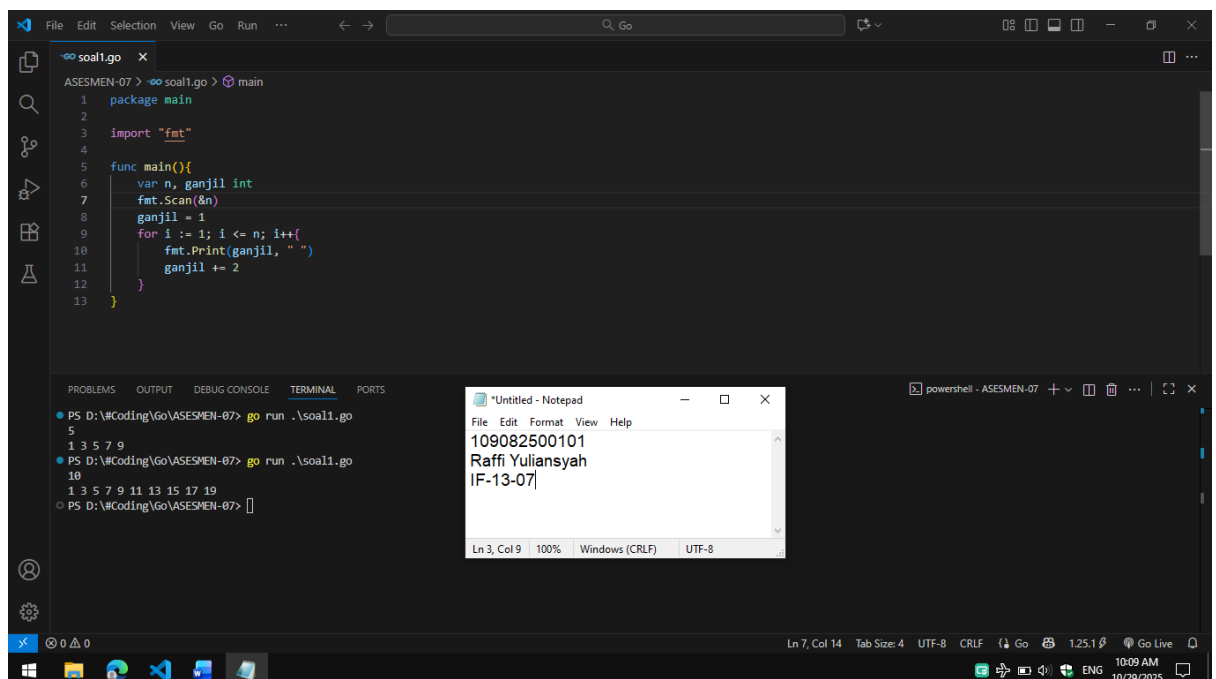
        fmt.Print(ganjil, " ")

        ganjil += 2

    }

}
```

Screenshoot program



Deskripsi program

Program tersebut adalah program yang dibuat untuk menampilkan deret bilangan ganjil sebanyak n.

Pada bagian awal program kita import "fmt" terlebih dahulu untuk I/O. kemudian didalam fungsi main(), akan dijalankan program utama. Pertama dideklarasikan dua variabel bertipe data integer, yaitu variabel n untuk menampung input banyak deret dan variabel ganjil untuk menyimpan bilangan ganjil. Setelah itu, program akan menerima input menggunakan fmt.Scan(&n) yang inputnya akan disimpan di variabel n. Karena bilangan ganjil dimulai dari 1, maka variabel ganjil diisikan dengan nilai 1 terlebih dahulu, ganjil = 1.

Untuk menampilkan banyak deret bilangan ganjil digunakan for-loop. for pada bagian awal berisi indeks (i) awal 1, i :=1, pada bagian kondisi, i <= n, dan i++. Artinya program tersebut akan menampilkan deret sebanyak 1 sampai dengan n. Pada bagian blok program for, untuk setiap iterasi nya kan melakukan : Pertama, program akan menampilkan isi dari variabel ganjil, fmt.Print(ganjil, " "). Kemudian variabel ganjil tersebut ditambah dengan 2, ganjil +=2 . Hal tersebut diulangi untuk setiap iterasi sampi kondisi bernilai salah dan program berhenti.

2. SOAL 2

Source Code

```
package main

import "fmt"

func main(){

    var start, end, hasil int

    fmt.Scan(&start)

    fmt.Scan(&end)

    hasil = 1

    for i := start; i <= end; i++){

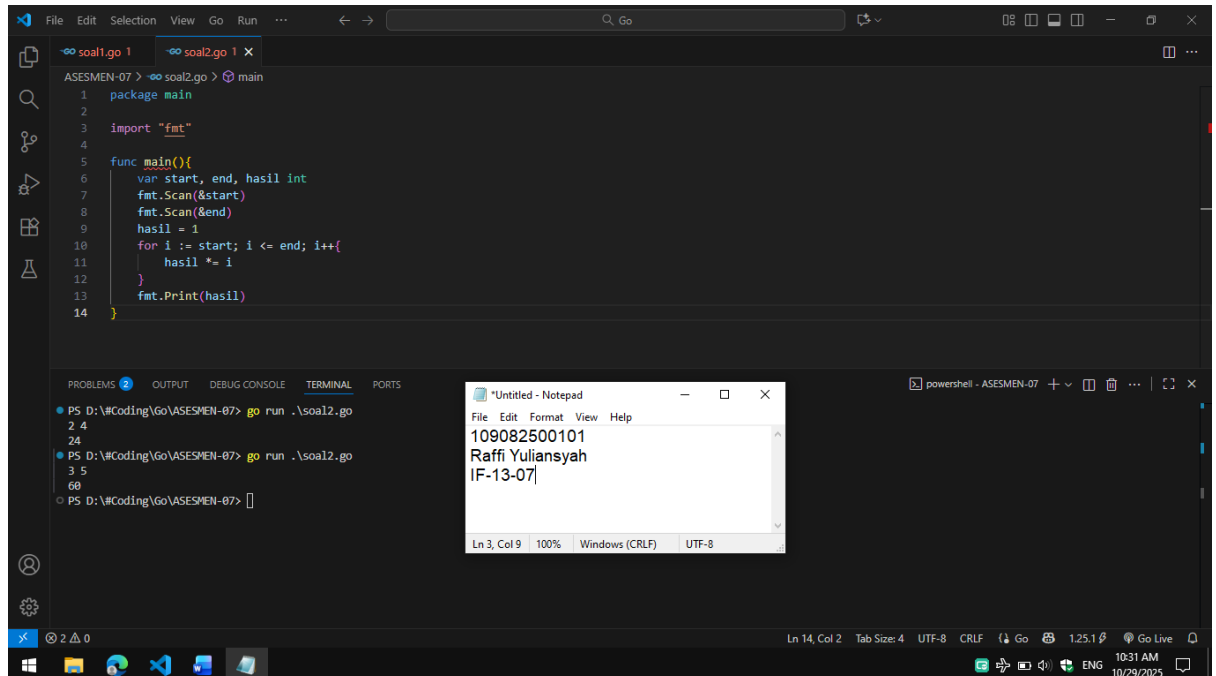
        hasil *= i

    }
```

```
fmt.Print(hasil)
```

```
}
```

Screenshoot program



Deskripsi program

Program ini bertujuan untuk menghitung hasil perkalian dari deret bilangan bulat berurut sesuai dengan rentang(awal sampai akhir) bilangan yang diberikan.

Pada bagian awal program kita import "fmt" terlebih dahulu untuk I/O. Kemudian didalam fungsi main(), akan dijalankan program utama. Pertama dideklarasikan tiga variabel bertipe data integer, yaitu variabel awal untuk menyimpan nilai awal nilai rentang, variabel akhir untuk menyimpan nilai akhir rentang, dan hasil untuk menyimpan hasil perkalian deret bilangan dari dari rentang yang diberikan. Setelah itu, pertama program akan diinputkan angka/nilai yang akan disimpan di variabel awal dan kedua program akan diinputkan angka/nilai yang akan disimpan di variabel akhir. Karena hasil ini akan menyimpan perkalian maka diberi nilai default 1 agar tidak nol. Untuk menghitung perkalian deret bilangan dalam rentang tertentu digunakan for-loop. for i := start; i <= end; i++, memiliki arti pengulangan akan dijalankan dari start (i) sampai dengan end secara berurutan. Pada bagian dalam for, pada setiap iterasinya program akan: mengalikan variabel hasil dengan i, hasil *= i. Karena i (nilai awal rentang) akan bertambah 1 setiap iterasinya maka hasil akan memiliki nilai dari i*(i+1)*(i+2)*... sampai rentang yang diberikan (hingga kondisi bernilai false). Kemudian pada bagian akhir ditampilkan variabel hasil yang berisi perkalian deret.

3. SOAL 3

Source Code

```

package main

import "fmt"

func main() {

    var keping, ikat, karung, peti int

    fmt.Scan(&keping)

    peti = keping / 800

    karung = keping % 800 / 100

    ikat = keping % 800 % 100 / 10

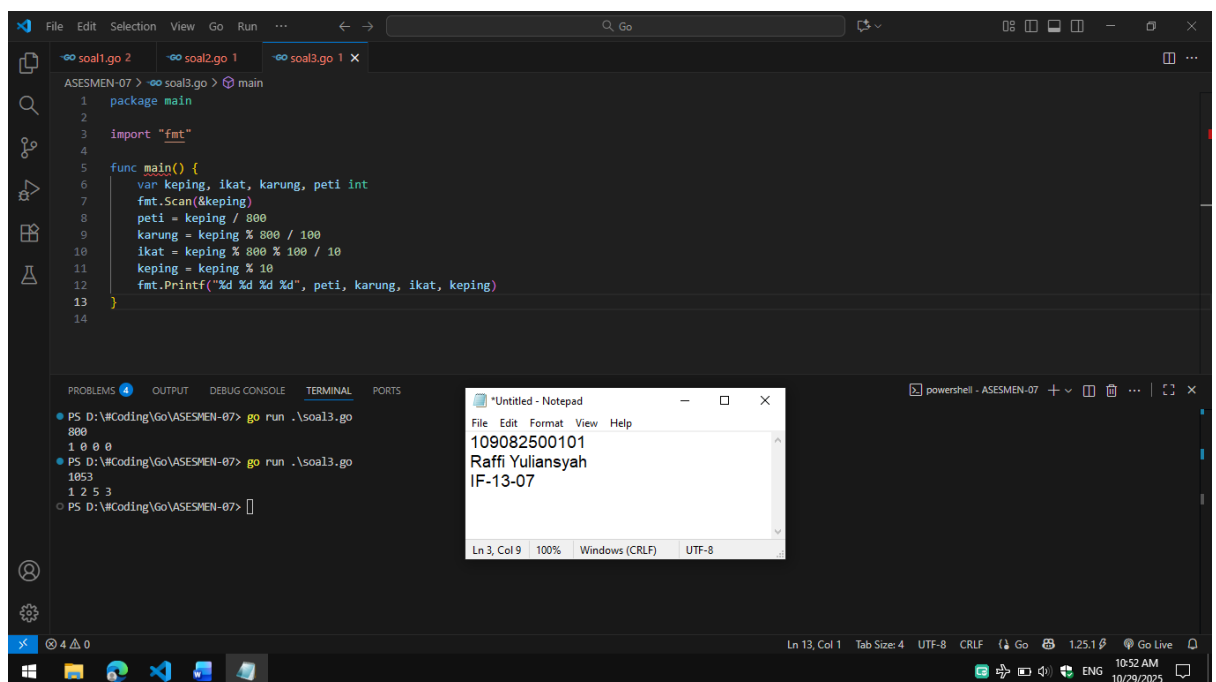
    keping = keping % 10

    fmt.Printf("%d %d %d %d", peti, karung, ikat,
keping)

}

```

Screenshoot program



Deskripsi program

Program ini bertujuan untuk mengonversi nilai keping ke satuan ikat, karung, dan peti. Pada bagian awal program kita import "fmt" terlebih dahulu untuk I/O. kemudian didalam fungsi main(), akan dijalankan program utama. Pertama dideklarasikan empat variabel bertipe data integer, yaitu keping untuk menyimpan nilai keping, ikat untuk menyimpan nilai ikat, karung untuk menyimpan nilai karung, dan peti untuk menyimpan nilai peti. Program akan diinputkan nilai yang akan disimpan ke variabel keping. Aturan konversi sebagai berikut.

1 peti = 8 karung, 1 karung = 10 ikat, 10 ikat = 10 keping.

Dari aturan tersebut dapat ditranslasikan kedalam program dalam kode sebagai berikut.

```
peti = keping / 800

karung = keping % 800 / 100

ikat = keping % 800 % 100 / 10

keping = keping % 10
```

Kode tersebut memiliki arti:

1 peti = 800 keping, 1 karung = 100 keping, dan 1 ikat = 10 keping. Sisa keping yang tidak bisa dikonversi akan disimpan di variabel keping. Dalam proses konversi apabila nilai keping sudah dikonversi kesatuan tertentu maka konversi selanjutnya akan menggunakan sisa dari keping konversi sebelumnya.

Setelah variabel tersebut dikonversi kesetiap satuan dan disimpan di masing-masing variabel, program akan menampilkan hasil konversi dengan urutan (peti) (karung) (ikat) (keping).