

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

MANGGALA PATRA RADITYA

109082500179

S1IF-13-02

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

SOAL

1. SOAL 1

Source Code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukkan jumlah bilangan ganjil: ")

    fmt.Scan(&n)

    fmt.Print("Output: ")

    for i := 1; i <= n; i++ {
        fmt.Print(2*i - 1)

        if i < n {
            fmt.Print(" ")
        }
    }

    fmt.Println()
}
```

Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan jumlah bilangan ganjil: ")
8     fmt.Scan(&n)
9
10    fmt.Print("Output: ")
11    for i := 1; i <= n; i++ {
12        fmt.Print(2*i - 1)
13        if i < n {
14            fmt.Print(" ")
15        }
16    }
17    fmt.Println()
18 }
```

Terminal Output:

```
PS C:\Users\User\OneDrive\Documents\Tipe Soal B> go run "C:\Users\User\OneDrive\Documents\Tipe Soal B\Soal1\main.go"
Masukkan jumlah bilangan ganjil: 5
Output: 1 3 5 7 9
PS C:\Users\User\OneDrive\Documents\Tipe Soal B>
```

Deskripsi program

Program di atas merupakan program sederhana dalam bahasa **Go (Golang)** yang berfungsi untuk menampilkan **deret bilangan ganjil** sebanyak jumlah yang dimasukkan oleh pengguna.

Pertama, program meminta pengguna untuk memasukkan sebuah angka n melalui perintah `fmt.Scan(&n)`. Angka ini menunjukkan berapa banyak bilangan ganjil yang ingin ditampilkan. Setelah itu, program mencetak teks "Output: " sebagai penanda hasil keluaran.

Bagian utama dari program terletak pada perulangan `for i := 1; i <= n; i++`. Di dalam perulangan tersebut, program menghitung bilangan ganjil ke- i dengan rumus $2*i - 1$. Rumus ini menghasilkan deret bilangan ganjil seperti 1, 3, 5, 7, 9, dan seterusnya. Agar hasilnya rapi, program juga menambahkan spasi antar angka menggunakan `if i < n` supaya tidak ada spasi di akhir deret.

Setelah semua bilangan ganjil dicetak, program menutup baris dengan `fmt.Println()` agar hasil output tampil di satu baris yang bersih.

Secara keseluruhan, program ini membantu menampilkan deret bilangan ganjil pertama secara berurutan mulai dari 1, sesuai jumlah yang diinginkan pengguna, dengan tampilan yang sederhana dan mudah dipahami.

2. SOAL 2

Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Scan(&x, &y)

    jumlahBakteri := 1

    for i := x; i <= y; i++ {

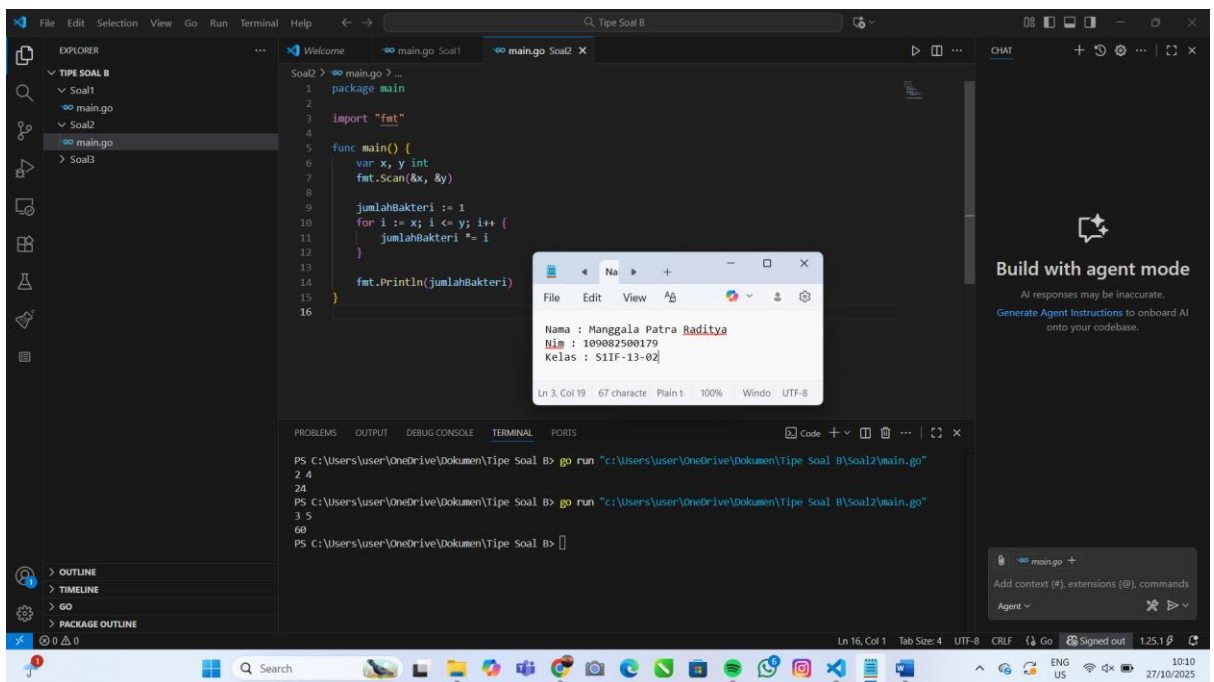
        jumlahBakteri *= i

    }

    fmt.Println(jumlahBakteri)

}
```

Screenshoot program



```
Soal2 > main.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7     fmt.Scan(&x, &y)
8
9     jumlahBakteri := 1
10    for i := x; i <= y; i++ {
11        jumlahBakteri *= i
12    }
13
14    fmt.Println(jumlahBakteri)
15 }
16
```

```
PS C:\Users\user\OneDrive\Documents\Tipe Soal B> go run "c:\Users\user\OneDrive\Documents\Tipe Soal B\Soal2\main.go"
2 4
24
PS C:\Users\user\OneDrive\Documents\Tipe Soal B> go run "c:\Users\user\OneDrive\Documents\Tipe Soal B\Soal2\main.go"
3 5
60
PS C:\Users\user\OneDrive\Documents\Tipe Soal B>
```

Deskripsi program

Program di atas merupakan program sederhana dalam bahasa **Go (Golang)** yang digunakan untuk menghitung **pertumbuhan jumlah bakteri** dari hari ke-x sampai hari ke-y.

Cara kerjanya, program pertama-tama meminta dua buah input dari pengguna, yaitu nilai x (hari awal) dan y (hari akhir). Setelah itu, program mendefinisikan variabel jumlahBakteri dengan nilai awal 1, karena pada awalnya hanya ada satu bakteri.

Kemudian, menggunakan perulangan for $i := x; i \leq y; i++$, program akan mengalikan jumlahBakteri dengan nilai i pada setiap iterasi. Logikanya, setiap hari jumlah bakteri berkembang sebanyak kelipatan hari tersebut. Jadi misalnya jika pengguna memasukkan 2 dan 4, maka perhitungannya menjadi $2 \times 3 \times 4 = 24$.

Setelah perulangan selesai, program mencetak hasil akhir yang menunjukkan jumlah total bakteri pada hari terakhir (y).

Secara sederhana, program ini menggambarkan proses pertumbuhan bakteri yang berkembang setiap hari sesuai dengan urutan harinya, dan hasil akhirnya menunjukkan total bakteri setelah beberapa hari pertumbuhan.

3. SOAL 3

Source Code

```
package main

import "fmt"

func main() {

    var keping int

    fmt.Scan(&keping)

    peti := keping / 800

    sisa := keping % 800

    karung := sisa / 100

    sisa %= 100

    ikat := sisa / 10

    keping = sisa % 10

    fmt.Println(peti, "peti,", karung, "karung,",
    ikat, "ikat,", "dan", keping, "keping")

}
```

Screenshoot program

The screenshot shows a Visual Studio Code editor with a Go program. The code in `main.go` is as follows:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var keping int
7     fmt.Scan(&keping)
8
9     peti := keping / 800
10    sisa := keping % 800
11
12    karung := sisa / 100
13    sisa %= 100
14
15    ikat := sisa / 10
16    keping = sisa % 10
17
18    fmt.Println(peti, "peti,", karung, "karung,", ikat, "ikat,", "dan", keping, "keping")
19 }
```

A small input window is open, showing the following text:

```
Nama : Manggala Patra Raditya
Nim : 109082500179
Kelas : SIIF-13-02
```

The terminal at the bottom shows the execution of the program:

```
PS C:\Users\user\OneDrive\Dokumen\Tipe Soal B> go run "C:\Users\user\OneDrive\Dokumen\Tipe Soal B\Soal3\main.go"
800
1 peti, 0 karung, 0 ikat, dan 0 keping
PS C:\Users\user\OneDrive\Dokumen\Tipe Soal B> go run "C:\Users\user\OneDrive\Dokumen\Tipe Soal B\Soal3\main.go"
1053
1 peti, 2 karung, 5 ikat, dan 3 keping
PS C:\Users\user\OneDrive\Dokumen\Tipe Soal B>
```

Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go (Golang) yang digunakan untuk menghitung konversi satuan barang dagang di Jalur Sutra dari satuan keping menjadi peti, karung, ikat, dan sisa keping.

Cara kerjanya cukup sederhana. Program pertama-tama meminta pengguna memasukkan jumlah keping. Setelah itu, program menghitung berapa banyak satuan peti yang bisa dibentuk dari jumlah keping tersebut. Karena 1 peti setara dengan 800 keping, maka jumlah peti diperoleh dengan $\text{keping} / 800$. Sisa keping yang belum terpakai kemudian disimpan dalam variabel sisa.

Langkah berikutnya, program menghitung berapa banyak karung dari sisa keping tersebut dengan rumus $\text{sisa} / 100$, karena 1 karung = 100 keping. Setelah itu, sisa kembali dihitung untuk mengetahui berapa ikat ($\text{sisa} / 10$), dan terakhir sisanya lagi merupakan jumlah keping yang tersisa ($\text{sisa} \% 10$).

Terakhir, program mencetak hasilnya dalam format yang mudah dibaca, misalnya: "1 peti, 2 karung, 5 ikat, dan 3 keping."

Secara singkat, program ini membantu mengubah jumlah keping menjadi satuan yang lebih besar sesuai sistem perdagangan zaman Jalur Sutra.