

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

**MODUL 3 DAN 4
FUNGSI DAN PROSEDUR**



Oleh:

HANAH NUR AZIZAH

2311102312

IF-11-07

**PROGRAM STUDY S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO
2024**

I. DASAR TEORI

3.1 Definisi Function

Fungsi merupakan satu kesatuan rangkain instruksi yang memberikan atau menghasilkan suatu nilai dan biasanya memetakan input ke suatu nilai yang lain. Oleh karena itu, fungsi selalu menghasilkan/mengebalikan nilai. Suatu subprogram dikatakan fungsi apabila:

1. Ada deklarsi tipe nilai yang dikembalikan, dan
2. Terdapat kata kunci return dalam badan subprogram.

Maka fungsi digunakan jika suatu nilai biasanya diperlukan, seperti:

- Assignment nilai ke suatu variabel
- Bagian dari ekspresi
- Bagian dari argument suatu subprogram, dsb.

Karena itu pilih nama fungsi yang menggambarkan nilai, seperti kata benda dan kata sifat. Contoh nama-nama fungsi: median, rerata, nilaiTerbesar, ketemu, selesai, ...

3.2 Deklarasi Function

Deklarasi fungsi sama dengan prosedur, yaitu berada pada blok yang terpisah dengan program utama.

3.3 Cara Pemanggilan Function

Pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi beserta argument yang diminta oleh parameter dan fungsi

4.1 Definisi Prosedure

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi baru yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar. Prosedur akan menghasilkan suatu akibat atau efek langsung pada program Ketika dipanggil pada program utama. Suatu subprogram dikatakan prosedur apabila:

1. Tidak ada deklarasi tipe nilai yang dikembalikan, dan
2. Tidak terdapat kata kunci return dalam badan subprogram.

Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) dan/atau instruksi yang berasal dari paket (fmt), seperti fmt.Scan, dan fmt.Print. Karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur. Contoh: cetak, hitungRerata, cariNilai, belok, mulai,...

4.2 Deklarasi Procedure

Penulisan deklarasi ini berada di luar blok yang dari program utama atau func main() pada suatu program Go, dan bisa ditulis sebelum atau setelah dari blok program utama tersebut.

4.3 Cara Pemanggilan Procedure

Suatu prosedur hanya akan dieksekusi apabila dipanggil baik secara langsung atau tidak langsung oleh program utama. Tidak langsung di sini maksudnya adalah prosedur dipanggil oleh program utama melalui perantara subprogram yang lain.

Pemanggilan suatu procedure cukup mudah, yaitu dengan hanya menuliskan nama beserta parameter atau argument yang diminta dari suatu prosedur. Sebagai contoh prosedur cetakNFibo di atas dipanggil dengan menuliskan Namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argument untuk parameter n.

4.4 Parameter

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilannya melalui argument yang diberikan melalui parameter yang dideklarasikan pada subprogramnya. Berikut ini jenis atau pembagian dari parameter.

Berdasarkan letak penulisannya pada program, maka parameter dapat dikelompokkan menjadi dua, yaitu:

1. Parameter Formal

Parameter formal adalah parameter yang ditulis pada saat deklarasi suatu subprogram, parameter ini berfungsi sebagai petunjuk bahwa argument apa saja yang diperlukan pada saat pemanggilan subprogram.

2. Parameter Aktual

Sedangkan parameter actual adalah argument yang digunakan pada parameter saat pemanggilan suatu subprogram. Banyaknya argument

dan tipe data yang terdapat pada parameter actual harus mengikuti parameter formal.

Selain itu parameter juga dikelompokkan berdasarkan alokasi memorinya, yaitu:

1. Pass by Value

Nilai pada parameter aktual akan disalin ke variabel lokal (parameter formal) pada subprogram. Artinya parameter aktual dan formal dialokasikan di dalam memori komputer dengan alamat memori yang berbeda. Subprogram dapat menggunakan nilai pada parameter formal tersebut untuk proses apapun, tetapi tidak dapat mengembalikan informasinya ke pemanggil melalui parameter aktual karena pemanggil tidak dapat mengakses memori yang digunakan oleh subprogram. Pass by value bisa digunakan baik oleh fungsi ataupun prosedur. Pada notasi pseudocode, secara semua parameter formal pada fungsi adalah pass by value, sedangkan pada prosedur diberi kata kunci in pada saat penulisan parameter formal. Sedangkan pada bahasa pemrograman Go sama seperti fungsi pada pseudocode, tidak terdapat kata kunci khusus untuk parameter formal fungsi dan prosedur.

2. Pass by Reference (Pointer)

Ketika parameter didefinisikan sebagai pass by reference, maka pada saat pemanggilan parameter formal akan berperan sebagai pointer yang menyimpan alamat memori dari parameter aktual. Sehingga perubahan nilai yang terjadi pada parameter formal tersebut akan berdampak pada parameter aktual. Artinya nilai terakhirnya akan dapat diketahui oleh si pemanggil informatika setelah subprogram tersebut selesai dieksekusi. Pass by reference sebaiknya digunakan hanya untuk prosedur. Penulisan parameter pass by reference pada prosedur baik pseudocode dan Go menggunakan kata kunci atau identifier khusus. Pada pseudocode menggunakan kata kunci in/out, sedangkan pada bahasa Go diberi identifier asterik (*) sebelum tipe data di parameter formal yang menjadi pass by reference.

II. GUIDED

1. Source Code

```
package main

import (
    "fmt"
)

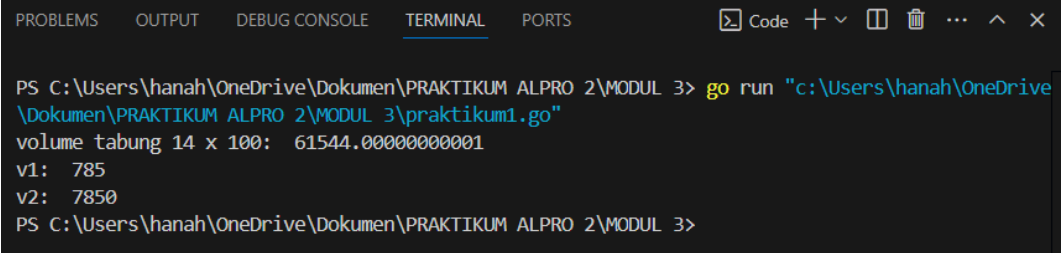
func volumeTabung(jari_jari, tinggi int) float64{
    var luasAlas, volume float64
    luasAlas = 3.14 *float64(jari_jari*jari_jari)
    volume = luasAlas * float64(tinggi)
    return volume
}

func main(){
    var r,t int
    var v1,v2 float64
    r = 5
    t = 10

    v1 = volumeTabung(r,t)
    v2 = volumeTabung(r,t) + volumeTabung(15,t)

    fmt.Println("volume tabung 14 x 100: ", volumeTabung(14,100))
    fmt.Println("v1: ",v1)
    fmt.Println("v2: ",v2)
}
```

Screenshot Output



```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c:\Users\hanah\OneDrive\
Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\praktikum1.go"
volume tabung 14 x 100:  61544.00000000001
v1:  785
v2:  7850
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3>
```

Penjelasan

Program di atas adalah program Go yang menghitung volume tabung berdasarkan jari-jari dan tinggi yang diberikan. Fungsi volumeTabung menerima dua parameter (jari_jari dan tinggi) dan mengembalikan nilai volume dalam tipe float64. Volume dihitung dengan rumus volume tabung, di mana r adalah jari-jari dan t adalah tinggi tabung, dengan nilai π didekati menjadi 3.14. Dalam fungsi main, variabel r dan t

diinisialkan dengan nilai 5 dan 10, yang digunakan untuk menghitung volume tabung pertama (v1). Volume tabung kedua (v2) dihitung dengan menjumlahkan volume tabung berukuran $r = 5$ dan $t = 10$ dengan volume tabung yang memiliki jari-jari 15 dan tinggi yang sama (10). Program juga mencetak volume tabung dengan jari-jari 14 dan tinggi 100, serta nilai v1 dan v2 menggunakan `fmt.Println`.

2. Source Code

```
package main

import (
    "fmt"
)

func cetakNFibo(n int){
    var f1,f2,f3 int
    f2 = 0;
    f3 = 1
    for i:= 1; i<=n; i++ {
        fmt.Println(f3)
        f1 = f2
        f2 = f3
        f3 = f1 + f2
    }
}

func main(){
    var x int
    x = 5
    cetakNFibo(x)
    cetakNFibo(100)
}
```

Screenshot Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code + - [] [X] ... ^ X

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c:\Users\hanah\OneDrive
\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\praktikum2.go"
1
1
2
3
5
1
1
2
3
```

Penjelasan

Program di atas adalah program Go yang menentukan apakah suatu tahun adalah tahun kabisat atau tidak. Pertama, program meminta input tahun dari pengguna dan menyimpannya dalam variabel tahun.

Kemudian, program memanggil fungsi cekKabisat() yang menerima tahun tersebut sebagai argumen dan mengembalikan nilai boolean (true jika kabisat, false jika bukan). Fungsi ini menggunakan aturan tahun kabisat: jika tahun habis dibagi 400, maka kabisat; jika habis dibagi 100, bukan kabisat; jika habis dibagi 4, maka kabisat. Hasilnya dicetak dengan label "Kabisat: ".

III. UNGUIDED

1. Minggu ini, mahasiswa Fakultas Informatika mendapat tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikan ke dalam suatu program. Oleh karena itu bersedia kalian membantu Jonas?

Masukan terdiri dari empat buah bilangan asli a , b , c , dan d yang dipisahkan oleh spasi, dengan syarat $a \geq c$ dan $b \geq d$.

Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap c , sedangkan baris kedua adalah hasil permutasi dan kombinasi b terhadap d .

Catatan: permutasi (P) dan kombinasi (C) dari n terhadap r ($n \geq r$) dapat dihitung dengan menggunakan persamaan berikut!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

No	Masukan	Keluaran	Penjelasan
1	5 10 3 10	60 10 3628800 1	$P(5, 3) = 5! / 2! = 120 / 2 = 60$ $C(5, 3) = 5! / (3! \times 2!) = 120 / 12 = 10$ $P(10, 10) = 10! / 0! = 3628800 / 1 = 3628800$ $C(10, 10) = 10! / (10! \times 0!) = 10! / 10! = 1$
2	8 0 2 0	56 28 1 1	

Selesaikan program tersebut dengan memanfaatkan subprogram yang diberikan berikut ini!

```
function factorial(n: integer) → integer
{mengembalikan nilai faktorial dari n}

function permutation(n, r : integer) → integer
{Mengembalikan hasil n permutasi r, dan n >= r}

function combination(n, r : integer) → integer
{Mengembalikan hasil n kombinasi r, dan n >= r}
```

Source Code

```
package main

import (
    "fmt"
)

//HANAH NUR AZIZAH 2311102312
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    return n * factorial(n-1)
}

func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    var a, b, c, d int
    fmt.Scan(&a, &b, &c, &d)

    p_ac := permutation(a, c)
    c_ac := combination(a, c)

    p_bd := permutation(b, d)
    c_bd := combination(b, d)

    fmt.Printf("No\tMasukan\t\t\tKeluaran\n")
    fmt.Printf("1\t%d %d %d %d\t\t%d\t%d\n", a, b, c, d, p_ac,
c_ac)
    fmt.Printf("2\t%d %d %d %d\t\t%d\t%d\n", b, d, b, d, p_bd,
c_bd)
}
```

Screenshot Output

```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c
:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\unguided3.1.go"
8 0 2 0
No      Masukan                               Keluaran
1       8 0 2 0                56         28
2       0 0 0 0                1          1
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> 
```


Penjelasan

Program di atas adalah program Go yang menghitung permutasi dan kombinasi menggunakan konsep faktorial. Fungsi factorial menghitung faktorial dari bilangan n menggunakan rekursi, di mana jika n bernilai 0, hasilnya adalah 1, dan jika tidak, nilai faktorial dihitung dengan mengalikan n dengan faktorial dari $n-1$. Dalam fungsi main, pengguna memasukkan empat angka (a , b , c , dan d). Program kemudian menghitung permutasi dan kombinasi untuk dua pasangan bilangan: pertama, menghitung permutasi dan kombinasi dari a dan c , lalu dari b dan d . Hasilnya ditampilkan dalam format tabel, di mana baris pertama mencetak permutasi dan kombinasi untuk nilai a dan c , dan baris kedua untuk nilai b dan d .

2. Diberikan tiga buah fungsi matematika yaitu $f(x) = x^2$, $g(x) = x - 2$ dan $h(x) = x + 1$. Fungsi komposisi (fogoh) (x) artinya adalah $f(g(h(x)))$. Tuliskan $f(x)$, $g(x)$ dan $h(x)$ dalam bentuk function.
- Masukan** terdiri dari sebuah bilangan bulat a , b dan c yang dipisahkan oleh spasi.
- Keluaran** terdiri dari tiga baris. Baris pertama adalah (fogoh) (a), baris kedua (gohof) (b), dan baris ketiga adalah (hofog) (c)!

No	Masukan	Keluaran	Penjelasan
1	7 2 10	36 3 65	(fogog)(7) = 36 (gohof)(2) = 3 (hofog)(10) = 65
2	5 5 5	16 24 10	(fogog)(5) = 16 (gohof)(5) = 24 (hofog)(5) = 10
3	3 8 4	4 63 5	(fogog)(5) = 4 (gohof)(5) = 63 (hofog)(5) = 5

Source Code

```
package main

import (
    "fmt"
)

//HANAH NUR AZIZAH 2311102312
func f(x int) int {
    return x * x
}

func g(x int) int {
```

```
        return x - 2
    }

    func h(x int) int {
        return x + 1
    }

    func fogoh(x int) int {
        return f(g(h(x)))
    }

    func gohof(x int) int {
        return g(h(f(x)))
    }

    func hofog(x int) int {
        return h(f(g(x)))
    }

    func main() {
        var a, b, c int

        fmt.Println("Masukan: ")
        fmt.Scan(&a, &b, &c)

        fogoh_a := fogoh(a)
        gohof_b := gohof(b)
        hofog_c := hofog(c)

        fmt.Println("\nKeluaran:")
        fmt.Println(fogoh_a)
        fmt.Println(gohof_b)
        fmt.Println(hofog_c)

        fmt.Println("\nPenjelasan:")
        fmt.Printf("(fogoh)(%d) = %d\n", a, fogoh_a)
        fmt.Printf("(gohof)(%d) = %d\n", b, gohof_b)
        fmt.Printf("(hofog)(%d) = %d\n", c, hofog_c)
    }
}
```

Screenshot Output

```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c:\Users\hanah\OneDrive
\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\unguided3.2.go"
Masukan:
7 2 10

Keluaran:
36
3
65

Penjelasan:
(fogoh)(7) = 36
(gohof)(2) = 3
(hofog)(10) = 65
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> █
```

Penjelasan

Program ini merupakan sebuah program Go yang mendefinisikan beberapa fungsi matematika sederhana dan menggabungkannya dalam komposisi fungsi. Ada tiga fungsi dasar: $f(x)$ yang mengkuadratkan input, $g(x)$ yang mengurangi input dengan 2, dan $h(x)$ yang menambahkan 1 pada input. Fungsi-fungsi gabungan seperti $fogoh(x)$, $gohof(x)$, dan $hofog(x)$ dibentuk dengan menggabungkan pemanggilan dari fungsi dasar tadi. Pada fungsi main, program meminta pengguna memasukkan tiga bilangan, lalu menghitung hasil dari $fogoh(a)$, $gohof(b)$, dan $hofog(c)$ berdasarkan masukan. Hasil dan penjelasan perhitungan ditampilkan pada output.

3. [**Lingkaran**] Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r . Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (x, y) berdasarkan dua lingkaran tersebut.

Masukan terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat.

Keluaran berupa string yang menyatakan posisi titik "**Titik di dalam lingkaran 1 dan 2**", "**Titik di dalam lingkaran 1**", "**Titik di dalam lingkaran 2**", atau "**Titik di luar lingkaran 1 dan 2**".

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3	Titik di dalam lingkaran 2

	4 5 6 7 8	
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

Fungsi untuk menghitung jarak titik (a, b) dan (c, d) dimana rumus jarak adalah:

$$jarak = \sqrt{(a - c)^2 + (b - d)^2}$$

dan juga fungsi untuk menentukan posisi sebuah titik sembarang berada di dalam suatu lingkaran atau tidak.

```
function jarak(a,b,c,d : real) -> real
{Mengembalikan jarak antara titik (a,b) dan titik (c,d)}

function didalam(cx,cy,r,x,y : real) -> boolean
{Mengembalikan true apabila titik (x,y) berada di dalam lingkaran yang
memiliki titik pusat (cx,cy) dan radius r}
```

Source Code

```
package main

import (
    "fmt"
    "math"
)

//HANAH NUR AZIZAH 2311102312
func hitungJarak(x1, y1, x2, y2 int) float64 {
    return math.Sqrt(math.Pow(float64(x2-x1), 2) +
math.Pow(float64(y2-y1), 2))
}

func diDalamLingkaran(cx, cy, r, x, y int) bool {
    return hitungJarak(cx, cy, x, y) <= float64(r)
}

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y int
    fmt.Println("Masukkan koordinat pusat lingkaran 1 (cx1, cy1) dan
jari-jari (r1):")
    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Println("Masukkan koordinat pusat lingkaran 2 (cx2, cy2) dan
```

```

fmt.Scan(&cx2, &cy2, &r2)

fmt.Println("Masukkan koordinat titik (x, y):")
fmt.Scan(&x, &y)

diLingkaran1 := diDalamLingkaran(cx1, cy1, r1, x, y)
diLingkaran2 := diDalamLingkaran(cx2, cy2, r2, x, y)

fmt.Println("\nNo\tMasukan\t\t\tKeluaran")
fmt.Println("-----")
fmt.Printf("1\t%d %d %d\n", cx1, cy1, r1)
fmt.Printf("\t%d %d %d\n", cx2, cy2, r2)
fmt.Printf("\t%d %d\n", x, y)

if diLingkaran1 && diLingkaran2 {
    fmt.Println("\tTitik di dalam lingkaran 1 dan 2")
} else if diLingkaran1 {
    fmt.Println("\tTitik di dalam lingkaran 1")
} else if diLingkaran2 {
    fmt.Println("\tTitik di dalam lingkaran 2")
} else {
    fmt.Println("\tTitik di luar lingkaran 1 dan 2")
}
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c:\Users\hanah\OneDrive
\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\unguided3.3.go"
Masukkan koordinat pusat lingkaran 1 (cx1, cy1) dan jari-jari (r1):
1 2 3
Masukkan koordinat pusat lingkaran 2 (cx2, cy2) dan jari-jari (r2):
4 5 6
Masukkan koordinat titik (x, y):
7 8

No      Masukan      Keluaran
-----
1       1 2 3
        4 5 6
        7 8
        Titik di dalam lingkaran 2
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> 

```

Penjelasan

Kode ini merupakan program untuk menentukan apakah sebuah titik berada di dalam dua lingkaran. Program dimulai dengan mendefinisikan fungsi hitungJarak yang menghitung jarak antara dua titik menggunakan rumus jarak Euclidean. Fungsi diDalamLingkaran digunakan untuk memeriksa apakah sebuah titik berada di dalam lingkaran, dengan membandingkan jarak titik dari pusat lingkaran

dengan jari-jarinya. Di bagian utama (main), pengguna diminta untuk memasukkan koordinat pusat dan jari-jari dua lingkaran, serta koordinat titik yang akan diperiksa. Program kemudian menentukan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran dan menampilkan hasil dalam format tabel.

4. Minggu ini, mahasiswa Fakultas Informatika mendapat tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikan ke dalam suatu program. Oleh karena itu bersedia kalia membantu Jonas?

Masukan terdiri dari empat buah bilangan asli a , b , c , dan d yang dipisahkan oleh spasi, dengan syarat $a \geq c$ dan $b \geq d$.

Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap c , sedangkan baris kedua adalah hasil permutasi dan kombinasi b terhadap d .

Catatan: permutasi (P) dan kombinasi (C) dari n terhadap r ($n \geq r$) dapat dihitung dengan menggunakan persamaan berikut!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

Contoh

No	Masukan	Keluaran	Penjelasan
1	5 10 3 10	60 10 3628800 1	$P(5, 3) = 5! / 2! = 120 / 2 = 60$ $C(5, 3) = 5! / (3! \times 2!) = 120 / 12 = 10$ $P(10, 10) = 10! / 0! = 3628800 / 1 = 3628800$ $C(10, 10) = 10! / (10! \times 0!) = 10! / 10! = 1$
2	8 0 2 0	56 28 1 1	

Selesaikan program tersebut dengan memanfaatkan prosedur yang diberikan berikut ini!

```

procedure factorial(in n: integer, in/out hasil:integer)
{I.S. terdefinisi bilangan bulat positif n
 F.S. hasil berisi nilai faktorial dari n}

procedure permutation(in n,r : integer, in/out hasil:integer)
{I.S. terdefinisi bilangan bulat positif n dan r, dan n >= r
 F.S. hasil berisi nilai dari n permutasi r}

procedure combination(in n,r : integer, in/out hasil:integer)
{I.S. terdefinisi bilangan bulat positif n dan r, dan n >= r
 F.S. hasil berisi nilai dari n kombinasi r}

```

Source Code

```
package main

import (
    "fmt"
)

//HANAH NUR AZIZAH 2311102312
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    return n * factorial(n-1)
}

func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    var a, b, c, d int

    fmt.Println("Masukkan 4 bilangan bulat (a b c d):")
    fmt.Scan(&a, &b, &c, &d)

    p_ac := permutation(a, c)
    c_ac := combination(a, c)

    p_bd := permutation(b, d)
    c_bd := combination(b, d)

    fmt.Println("\nMasukan:")
    fmt.Printf("%d %d %d %d\n", a, b, c, d)

    fmt.Println("\nKeluaran:")
    fmt.Printf("%d %d\n", p_ac, c_ac)
    fmt.Printf("%d %d\n", p_bd, c_bd)

    fmt.Println("\nPenjelasan:")
    fmt.Printf("P(%d,%d) = %d!/(%d)! = %d/%d = %d\n", a, c, a, a-c,
factorial(a), factorial(a-c), p_ac)
```

```

    fmt.Printf("C(%d,%d) = %d!/(%d! x %d!) = %d/(%d x %d) = %d\n\n",
a, c, a, c, a-c, factorial(a), factorial(c), factorial(a-c), c_ac)

    fmt.Printf("P(%d,%d) = %d!/(%d)! = %d/%d = %d\n", b, d, b, b-d,
factorial(b), factorial(b-d), p_bd)
    fmt.Printf("C(%d,%d) = %d!/(%d! x %d!) = %d/(%d x %d) = %d\n",
b, d, b, d, b-d, factorial(b), factorial(d), factorial(b-d), c_bd)
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c:\Users\hanah\OneDrive
\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\unguided4.1.go"
\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\unguided4.1.go"
Masukkan 4 bilangan bulat (a b c d):
5 10 3 10

Masukan:
5 10 3 10

Keluaran:
60 10
3628800 1

Penjelasan:
P(5,3) = 5!/(2)! = 120/2 = 60
C(5,3) = 5!/(3! x 2!) = 120/(6 x 2) = 10

P(10,10) = 10!/(0)! = 3628800/1 = 3628800
C(10,10) = 10!/(10! x 0!) = 3628800/(3628800 x 1) = 1
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3>

```

Penjelasan

Program di atas merupakan program Go yang menghitung permutasi dan kombinasi dari empat bilangan bulat yang dimasukkan oleh pengguna. Fungsi factorial digunakan untuk menghitung faktorial suatu bilangan secara rekursif. Fungsi permutation menghitung permutasi dari dua bilangan dengan rumus $P(n,r) = n!/(n-r)!$, sedangkan fungsi combination menghitung kombinasi dengan rumus $C(n,r) = n!/r!(n-r)!$. Pada fungsi utama, pengguna diminta untuk memasukkan empat bilangan bulat (a, b, c, d), kemudian program menghitung permutasi dan kombinasi untuk pasangan bilangan (a, c) dan (b, d). Hasilnya ditampilkan dalam tiga bagian: masukan, keluaran, dan penjelasan perhitungan yang detail sesuai dengan rumus yang digunakan.

5. Kompetisi pemrograman Tingkat nasional berlangsung ketat. Setiap peserta diberikan 8 soal yang harus dapat diselesaikan dalam waktu 5

jam saja. Peserta yang berhasil menyelesaikan soal paling banyak dalam waktu paling singkat adalah pemenangnya.

Buat program gema yang mencari pemenang dari daftar peserta yang diberikan. Program harus dibuat modular, yaitu dengan membuat prosedur `hitungSkor` yang mengembalikan total soal dan total skor yang dikerjakan oleh seorang peserta, melalui parameter formal. Pembacaan nama peserta dilakukan di program utama, sedangkan waktu pengerjaan dibaca di dalam prosedur.

prosedure `hitungSkor(in/out soal, skor : integer)`

Setiap baris **masukan** dimulai dengan satu string nama peserta tersebut diikuti dengan adalah 8 integer yang menyatakan berapa lama (dalam menit) peserta tersebut menyelesaikan soal. Jika tidak berhasil atau tidak mengirimkan jawaban maka otomatis dianggap menyelesaikan dalam waktu 5 jam 1 menit (301 menit).

Satu baris **keluaran** berisi nama pemenang, jumlah soal yang diselesaikan, dan nilai yang diperoleh. Nilai adalah total waktu yang dibutuhkan untuk menyelesaikan soal yang berhasil diselesaikan.

No	Masukan	Keluaran
1	Astuti 20 50 301 301 61 71 75 10 Bertha 25 47 301 26 50 60 65 21 Selesai	Bertha 7 294

Keterangan:

Astuti menyelesaikan 6 soal dalam waktu 287 menit, sedangkan Bertha 7 soal dalam waktu 294 menit. Karena Bertha menyelesaikan lebih banyak, maka Bertha menang. Jika keduanya menyelesaikan sama banyak, maka pemenang adalah yang menyelesaikan dengan total waktu paling kecil.

Source Code

```
package main

import "fmt"
//HANAH NUR AZIZAH 2311102312
type Peserta struct {
    Nama      string
    Waktu     []int
    Skor      int
    Selesai   int
}

func hitungSkor(peserta *Peserta) {
    peserta.Skor = 0
```

```

peserta.Selesai = 0

for _, waktu := range peserta.Waktu {
    if waktu == 301 {
        continue
    } else {
        peserta.Skor += waktu
        peserta.Selesai++
    }
}

}

func main() {
    var peserta []Peserta

    fmt.Println("Masukan:")

    for i := 0; i < 2; i++ {
        var nama string
        var waktu [8]int

        fmt.Scan(&nama)
        for j := 0; j < 8; j++ {
            fmt.Scan(&waktu[j])
        }

        peserta = append(peserta, Peserta{
            Nama: nama,
            Waktu: waktu[:],
        })
    }

    fmt.Println("Selesai")

    for i := range peserta {
        hitungSkor(&peserta[i])
    }

    var pemenang Peserta
    for i := range peserta {
        if peserta[i].Selesai > pemenang.Selesai {
            pemenang = peserta[i]
        } else if peserta[i].Selesai == pemenang.Selesai {
            if pemenang.Skor == 0 || peserta[i].Skor < pemenang.Skor

```

```

        pemenang = peserta[i]
    }
}

fmt.Println("\nKeluaran:")
fmt.Printf("%s %d %d\n", pemenang.Nama, pemenang.Selesai,
pemenang.Skor)
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c:\Users\hanah\OneDrive
\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\unguided4.2.go"
Masukan:
astuti 20 50 301 301 61 71 75 10
bertha 25 47 301 26 50 60 65 21
Selesai

Keluaran:
bertha 7 294
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> 

```

Penjelasan

Program di atas menggunakan prosedur untuk menghitung skor dan menentukan pemenang dari sekumpulan peserta berdasarkan waktu pengerjaan. Tipe data Peserta menyimpan nama, waktu pengerjaan dalam bentuk slice integer, skor, dan jumlah tugas yang selesai. Prosedur hitungSkor bertugas menghitung total skor peserta dengan cara menjumlahkan setiap waktu yang tidak bernilai 301, serta menghitung jumlah tugas yang diselesaikan. Pada bagian utama program main, dua peserta dimasukkan melalui input, masing-masing dengan delapan waktu pengerjaan. Setelah input selesai, prosedur hitungSkor dipanggil untuk menghitung skor setiap peserta. Pemenang ditentukan berdasarkan jumlah tugas yang terselesaikan terbanyak. Jika jumlah tugas yang diselesaikan sama, peserta dengan skor terkecil dipilih sebagai pemenang. Hasil akhirnya menampilkan nama pemenang, jumlah tugas selesai, dan total skornya.

- Skiena dan Revilla dalam Programming Challenges mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat n . Jika bilangan n saat itu genap, maka suku berikutnya adalah $1/2n$, tetapi jika ganjil maka suku berikutnya bernilai $3n + 1$. Rumus yang sama digunakan terus menerus untuk mencari suku berikutnya. Deret berakhir ketika suku terakhir bernilai 1. Sebagai contoh jika dimulai dengan $n = 22$ maka deret bilangan yang diperoleh adalah:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Untuk suku awal sampai dengan 1000000, diketahui deret selalu mencapai suku dengan nilai 1.

Buat program **skiena** yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetak Deret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

```
procedure cetakDeret(in n : integer )
```

Masukan berupa satu bilangan integer positif yang lebih kecil dari 1000000.

Keluaran terdiri dari satu baris saja. Setiap suku dari deret tersebut dicetak dalam baris yang dan dipisahkan oleh sebuah spasi.

No	Masukan	Keluaran
1	22	22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Source Code

```
package main

import (
    "fmt"
)

//HANAH NUR AZIZAH 2311102312
func cetakDeret(n int) {
    for n != 1 {
        fmt.Printf("%d ", n)
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println(n)
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif kurang dari 1000000: ")
    fmt.Scan(&n)

    if n > 0 && n < 1000000 {
        cetakDeret(n)
    } else {
```

```
        fmt.Println("Bilangan harus positif dan kurang dari  
1000000")  
    }  
}
```

Screenshot Output

```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> go run "c:\Users\hanah\OneDrive  
\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3\unguided4.3.go"  
Masukkan bilangan bulat positif kurang dari 1000000: 22  
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1  
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 3> █
```

Penjelasan

Program di atas merupakan program Go untuk mencetak deret bilangan berdasarkan aturan tertentu. Prosedur bernama cetakDeret menerima parameter bilangan bulat positif n dan mencetak deret tersebut hingga mencapai angka 1. Prosedur ini menggunakan perulangan for yang terus berjalan selama nilai n tidak sama dengan 1. Jika n adalah bilangan genap, n dibagi 2, dan jika ganjil, n dikalikan 3 lalu ditambah 1. Pada bagian utama program main, pengguna diminta memasukkan bilangan bulat positif kurang dari 1.000.000. Jika input memenuhi syarat, prosedur cetakDeret dipanggil untuk memproses dan mencetak deret.

DAFTAR PUSTAKA

- [1] Modul 3: Fungsi
- [2] Modul 4: Prosedur