LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 4

PROSEDUR

Disusun Oleh :

Maulisa Elvita Sari / 2311102259

S1-IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Prosedur adalah sekumpulan pernyataan yang dapat dieksekusi untuk melakukan suatu tugas tertentu. Prosedur tidak mengembalikan nilai, meskipun dapat memiliki parameter untuk menerima input. Fungsi mirip dengan prosedur, tetapi dapat mengembalikan nilai. Dalam Go, fungsi adalah blok kode yang bisa dipanggil dengan nama tertentu.

Keuntungan Menggunakan Prosedur

Modularitas

Menggunakan prosedur memungkinkan programmer membagi logika bisnis besar menjadi potongan-potong kecil yang lebih mudah dikelola.

Reusabilitas

Suatu fungsi yang telah ditetapkan dapat digunakan berkali-kali di berbagai tempat dalam program tanpa harus menulis ulang kode. Hal ini efektif dalam menghemat waktu dan usaha saat melakukan iteratif development atau maintenance codebase.

Pemeliharaan

Memperbaiki bug atau menambah fitur baru menjadi lebih cepat ketika semua komponennya terisolasi dan terdefinisi dengan baik. Perubahan hanya perlu dilakukan di satu lokasi saja sehingga meminimalkan risiko

Perbedaan Fungsi Dan Prosedur

- **Fungsi**: Digunakan secara luas dalam program Go untuk mengemas kode yang berulang-ulang menjadi modul-modul yang lebih manejable. Contoh: Fungsi main() adalah fungsi utama yang dieksekusi ketika program dijalankan.
- **Prosedur**: Konsepnya sama dengan fungsi, tetapi istilah ini lebih umum digunakan dalam konteks non-teknis atau teoritis

II. GUIDED

1. Guided 1

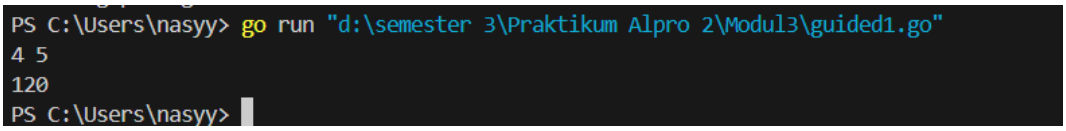
Soal Studi Case

Menghitung Faktorial dan Permutasi dalam prosedur

Sourcecode

```go
package main

import "fmt"


func main() {
    var a, b int
    fmt.Print("Masukkan dua bilangan: ")
    fmt.Scan(&a, &b)

    if a >= b {
        var hasil int
        permutasi(a, b, &hasil)
        fmt.Println("Hasil permutasi:", hasil)
    } else {
        var hasil int
        permutasi(b, a, &hasil)
        fmt.Println("Hasil permutasi:", hasil)
    }
}

func faktorial(n int, hasil *int) {
    *hasil = 1
    for i := 1; i <= n; i++ {
        *hasil *= i
    }
}

func permutasi(n, r int, hasil *int) {
    var faktN, faktNR int
    faktorial(n, &faktN)
    faktorial(n-r, &faktNR)
    *hasil = faktN / faktNR
}
```

Screenshoot Output



Deskripsi Program

Kode di atas adalah program Go yang digunakan untuk menghitung permutasi dua angka. Program meminta pengguna memasukkan dua bilangan, lalu menentukan bilangan mana yang lebih besar agar perhitungan permutasi berjalan dengan benar. Perhitungan dilakukan menggunakan prosedur, yaitu fungsi tanpa mengembalikan nilai, melainkan menyimpan hasilnya langsung ke variabel menggunakan pointer. Di dalam main, variabel hasil digunakan untuk menampung hasil perhitungan, yang kemudian dicetak dengan fmt.Println().

Prosedur faktorial menghitung faktorial dari bilangan n dan menyimpan hasilnya ke variabel yang dikirim melalui pointer (*int). Prosedur permutasi memanfaatkan prosedur faktorial untuk menghitung faktorial n dan (n-r), kemudian membagi keduanya untuk mendapatkan hasil permutasi. Dengan menggunakan pointer, hasil perhitungan langsung disimpan ke variabel yang digunakan di main, sehingga tidak perlu menggunakan return. Ini membuat kode lebih terstruktur dan memudahkan dalam mengelola hasil perhitungan.

2. Guided 2

Soal Studi Case

Menghitung luas dan keliling persegi dalam prosedur

Sourcecode

```go
package main

import "fmt"

func main() {
    var s int
    fmt.Print("Masukkan Sisi: ")
    fmt.Scan(&s)

    luasPersegi(s)
    kelilingPersegi(s)

}
func luasPersegi(s int) {
    hasil := s * s
    fmt.Println("Hasil Luas: ", hasil)
}

func kelilingPersegi(s int) {
    hasil := 4 * s
    fmt.Print("Hasil keliling: ", hasil)
}
```

Screenshoot Output

```
PS C:\Users\nasyy> go run "d:\semester 3\Praktikum Alpro 2\Modul4\guided2.go"
Masukkan Sisi: 5
Hasil Luas:  25
Hasil keliling: 20
PS C:\Users\nasyy>
```

Deskripsi Program

Kode di atas adalah program Go untuk menghitung luas dan keliling persegi. Program ini meminta pengguna memasukkan panjang sisi persegi, kemudian menghitung dan menampilkan hasilnya.

Prosedur luasPersegi digunakan untuk menghitung luas dengan rumus s * s, sedangkan kelilingPersegi menghitung keliling dengan rumus 4 * s. Kedua

hasil perhitungan langsung dicetak di layar menggunakan fmt.Println() dan fmt.Print().

I. UNGUIDED

1. Unguided 1

Soal Studi Case

Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalian membantu Jonas? (tidak tentunya ya :p) Masukan terdiri dari empat buah bilangan asli a, b, c, dan d yang dipisahkan oleh spasi, dengan syarat ac dan b ≥ 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≥r) dapat dihitung dengan menggunakan persamaan berikut! P(n,r) = n! / (n-r)!, sedangkan C(n,r) = n! / r!(n-r)!

Sourcecode

```go
package main

import "fmt"

// Annasya MZ_2311102246

func main() {
    var x1, x2, y1, y2 int

    fmt.Print("Masukkan empat bilangan: ")
    fmt.Scan(&x1, &x2, &y1, &y2)

    var perm1, komb1, perm2, komb2 int
    hitungPermutasi(x1, y1, &perm1)
    hitungKombinasi(x1, y1, &komb1)
    hitungPermutasi(x2, y2, &perm2)
    hitungKombinasi(x2, y2, &komb2)

    fmt.Println(perm1, komb1)
    fmt.Println(perm2, komb2)
}
```

```go
// Prosedur menghitung faktorial
func hitungFaktorial(n int, hasil *int) {
    *hasil = 1
    for i := 1; i <= n; i++ {
        *hasil *= i
    }
}

// Prosedur menghitung permutasi
func hitungPermutasi(n, r int, hasil *int) {
    var fN, fNR int
    hitungFaktorial(n, &fN)
    hitungFaktorial(n-r, &fNR)
    *hasil = fN / fNR
}

// Prosedur menghitung kombinasi
func hitungKombinasi(n, r int, hasil *int) {
    var fN, fR, fNR int
    hitungFaktorial(n, &fN)
    hitungFaktorial(r, &fR)
    hitungFaktorial(n-r, &fNR)
    *hasil = fN / (fR * fNR)
}
```

Screenshoot Output

```
PS C:\Users\nasyy> go run "d:\semester 3\Praktikum Alpro 2\Modul4\Unguided1.go"
Masukkan empat bilangan: 10 5 5 40
30240 252
120 0
PS C:\Users\nasyy>
```

Deskripsi Program

Program meminta pengguna untuk memasukkan empat bilangan bulat (x1, x2, y1, y2). Nilai-nilai tersebut akan digunakan untuk menghitung permutasi dan kombinasi secara berpasangan: (x1, y1) dan (x2, y2). Hasil perhitungan permutasi dan kombinasi ini disimpan dalam variabel perm1, komb1, perm2, dan komb2. Fungsi fmt.Scan digunakan untuk membaca input pengguna, dan hasil akhirnya ditampilkan dengan fmt.Println, yang mencetak permutasi dan kombinasi masing-masing pasangan.

Pada prosedur hitungFaktorial, perhitungan faktorial dilakukan dengan mengalikan angka dari 1 hingga n, dan hasilnya disimpan dalam variabel hasil menggunakan pointer. Prosedur ini mendukung fungsi hitungPermutasi dan hitungKombinasi, di mana faktorial digunakan sebagai bagian dari perhitungan rumus. Pada prosedur hitungPermutasi, rumus P(n, r) = n! / (n - r)! diterapkan, di mana faktorial n dan (n - r) dihitung terlebih dahulu. Demikian pula, prosedur hitungKombinasi menggunakan rumus C(n, r) = n! / (r! * (n - r)!) untuk menghitung kombinasi, dengan memanfaatkan faktorial n, r, dan (n - r). Setiap prosedur menggunakan pointer agar nilai hasil dapat diperbarui langsung di dalam fungsi pemanggilnya, meningkatkan efisiensi dan pengelolaan memori.

2. Unguided 2

Soal Studi Case

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

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 tau 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 nilal yang diperoleh. Nilai adalah total waktu yang dibutuhkan untuk menyelesaikan soal yang berhasil diselesaikan.

Sourcecode

```
package main

import "fmt"

func hitungSkor(waktu []int, soal *int, skor *int) {
```

```
    for _, durasi := range waktu {
        if durasi < 301 {
            *soal++         // Menambah jumlah soal yang
diselesaikan
            *skor += durasi // Menambah total waktu
        }
    }
}

func main() {
    var jumlahPeserta int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&jumlahPeserta)

    namaPemenang, soalTerbanyak, waktuTerbaik := "", -1,
99999

    for i := 0; i < jumlahPeserta; i++ {
        var namaPeserta string
        var waktuSoal [8]int
        fmt.Print("Masukkan nama dan waktu peserta: ")
        fmt.Scan(&namaPeserta, &waktuSoal[0], &waktuSoal[1],
&waktuSoal[2], &waktuSoal[3], &waktuSoal[4], &waktuSoal[5],
&waktuSoal[6], &waktuSoal[7])

        var totalSoal, totalWaktu int
        hitungSkor(waktuSoal[:], &totalSoal, &totalWaktu)

        if totalSoal > soalTerbanyak || (totalSoal ==
soalTerbanyak && totalWaktu < waktuTerbaik) {
            namaPemenang, soalTerbanyak, waktuTerbaik =
namaPeserta, totalSoal, totalWaktu
        }
    }

    fmt.Printf("Pemenang: %s\nJumlah soal: %d\nTotal waktu:
%d\n", namaPemenang, soalTerbanyak, waktuTerbaik)
}
```

Screenshoot Output

```
d:\semester 3\Praktikum Alpro 2\Modul4\unguided2.go:3:1: syntax error: imports must appear before other
PS C:\Users\nasyy> go run "d:\semester 3\Praktikum Alpro 2\Modul4\unguided2.go"
Masukan jumlah peserta: 2
Masukan nama dan waktu peserta: Astuti 20 50 301 301 61 71 75 10
Masukan nama dan waktu peserta: Bertha 25 47 301 26 50 60 65 21
Pemenang: Bertha
Jumlah soal: 7
Total waktu: 294
PS C:\Users\nasyy>
```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go untuk menentukan pemenang kompetisi penyelesaian soal. Dalam implementasinya, fungsi `hitungSkor` digunakan untuk mengolah data waktu yang dihabiskan oleh peserta untuk menyelesaikan 8 soal. Fungsi ini menerima dua parameter, yaitu slice dari waktu yang dihabiskan dan pointer untuk menyimpan jumlah soal yang diselesaikan serta total waktu. Di dalam fungsi, dilakukan iterasi terhadap setiap elemen dalam slice waktu, dan jika durasi kurang dari 301 detik, jumlah soal yang diselesaikan akan bertambah satu dan total waktu akan dijumlahkan.

Pada fungsi `main`, program meminta input dari pengguna untuk jumlah peserta serta waktu yang dihabiskan untuk setiap soal. Setelah itu, program memanggil `hitungSkor` untuk setiap peserta, menyimpan hasilnya, dan membandingkan jumlah soal yang diselesaikan serta waktu yang dihabiskan. Jika peserta baru menyelesaikan lebih banyak soal dibandingkan peserta sebelumnya, atau jika jumlah soal yang diselesaikan sama tetapi waktu totalnya lebih sedikit, maka peserta tersebut akan menjadi pemenang. Di akhir program, nama pemenang, jumlah soal yang diselesaikan, dan total waktu ditampilkan. Dengan struktur ini, program memberikan mekanisme yang jelas dan efisien untuk menentukan pemenang berdasarkan kriteria yang telah ditentukan.

3. Unguided 3

Soal Studi Case

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 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 sultu dari deret yang dijelaskan di atas untuk nilar suku awal yang diberihan. Pencetakan deret harus dibuat dalam prosedur cetak Deret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal. prosedure cetakberet (in n integer)

Masukan berupa satu bilangari integer positif yang lebih kecil dari 1000000.Keluaran terdiri dan satu baris saja. Setiap suhu carn deret tersebut dicetah dalam bans yang dan dipisahiran oleh sebuah spasi

Sourcecode

```go
package main

import "fmt"

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(1)
}

func main() {

    var n int
    fmt.Print("Masukkan nilai awal: ")
    fmt.Scan(&n)

    cetakDeret(n)
}
```

Screenshoot Output

```
PS C:\Users\nasyy> go run "d:\semester 3\Praktikum Alpro 2\Modul4\unguided3.go"
Masukkan nilai awal: 12
12 6 3 10 5 16 8 4 2 1
PS C:\Users\nasyy>
```

Deskripsi Program

Program ini dibuat untuk mencetak deret bilangan yang dimulai dari angka yang diberikan oleh pengguna, mengikuti aturan tertentu hingga mencapai angka 1. Jika angka yang diberikan adalah bilangan genap, maka angka tersebut dibagi dua, sedangkan jika angka tersebut adalah bilangan ganjil, maka dihitung dengan rumus `3*n + 1`. Proses ini diulang terus-menerus hingga angka menjadi 1, dan setiap nilai yang muncul selama proses ini akan dicetak dalam satu baris yang dipisahkan oleh spasi. Pada bagian utama program, pengguna diminta untuk memasukkan angka sebagai nilai awal. Setelah itu, program memanggil fungsi untuk mencetak deret sesuai dengan aturan yang dijelaskan. Program ini berjalan dengan menghitung setiap nilai bilangan hingga akhirnya mencapai angka 1, sesuai dengan rumus yang berlaku untuk bilangan genap dan ganjil.

Daftar Pustaka

https://www.detik.com/edu/detikpedia/d-5792538/procedure-text-struktur-ciri-dan-contohnya-dalam-bahasa-inggris
http://guntur.staff.gunadarma.ac.id/Publications/files/4644/GOLANG+FOR+BEGINNER+2018.pdf