LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

Modul 4 : Prosedur

Oleh:

ANGGUN WAHYU WIDIYANA

103112480280

12-IF-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

TAHUN AJARAN 2025/2026

I. DASAR TEORI

Prosedur adalah salah satu konsep fundamental dalam pemrograman yang digunakan untuk membuat program lebih modular, terstruktur, dan mudah dipelihara. Prosedur memungkinkan programmer untuk mengelompokkan instruksi atau kode yang sering digunakan menjadi satu kesatuan, sehingga dapat digunakan kembali di berbagai bagian program. Dalam algoritma dan pemrograman, prosedur merupakan subprogram yang tidak mengembalikan nilai, tetapi dapat melakukan tugas tertentu seperti mencetak data, melakukan perhitungan, atau memodifikasi variabel.

A. Definisi Prosedur

Prosedur adalah subprogram yang terdiri dari sekumpulan instruksi yang dikemas menjadi satu kesatuan untuk melakukan tugas tertentu. Prosedur tidak mengembalikan nilai (tidak ada return) seperti fungsi dalam pemrograman.

Ciri-ciri Prosedur:

1. Tidak memiliki tipe nilai yang dikembalikan.
2. Tidak menggunakan kata kunci return.
3. Dapat dipanggil dari program utama atau subprogram lain.
4. Nama prosedur biasanya berupa kata kerja (contoh: cetak, hitung, proses).

B. Kegunaan Prosedur

Prosedur memiliki berbagai kegunaan dalam pengembangan perangkat lunak:

1. Modularitas: Membagi program besar menjadi bagian-bagian kecil yang lebih mudah dikelola.
2. Reusabilitas: Kode prosedur dapat digunakan kembali tanpa perlu menulis ulang.
3. Efisiensi: Mengurangi duplikasi kode sehingga program lebih efisien.
4. Pemeliharaan: Mempermudah pembaruan dan debugging pada kode karena setiap bagian memiliki fungsi spesifik.

C. Struktur Deklarasi Prosedur

```
procedure namaProsedur(parameter1, parameter2, ...)
    // Deklarasi variabel lokal
    // Badan algoritma
endprocedure
```

Tabel 1.1 Notasi Psedeucode

```
```go
func namaProsedur(parameter1 tipe1, parameter2 tipe2) {
    // Deklarasi variabel lokal
    // Badan algoritma
}
```

Tabel 1.2 Notasi dalam Go
```

D. Cara Pemanggilan Prosedur

Pemanggilan prosedur dilakukan dengan menyebutkan nama prosedur beserta argumen sesuai parameter formal.

```
begin
    cetakPesan("Hello, World!")
end
```

Tabel 2.1 Notasi Psedeucode

```
```go
func main() {
    cetakPesan("Hello, World!")
}
```

Tabel 2.2 Notasi dalam Go

E. Parameter dalam Prosedur

Parameter adalah variabel yang digunakan untuk menerima nilai dari pemanggil prosedur. Parameter berdasarkan jenisnya yaitu

1. Parameter Formal: Parameter yang dideklarasikan di definisi prosedur.
   Contoh: panjang dan lebar pada deklarasi hitungLuas.
2. Parameter Aktual: Nilai yang diberikan saat pemanggilan prosedur.
   Contoh: hitungLuas(5, 10) → 5 dan 10 adalah parameter aktual.

```
```go
// Parameter formal: panjang dan lebar
func hitungLuas(panjang int, lebar int) int {
    return panjang * lebar
}

// Parameter aktual saat pemanggilan:
luas := hitungLuas(5, 10) // 5 dan 10 adalah parameter aktual
```

F. Kelebihan dan Kekurangan Prosedur

Kelebihan: Membantu modularisasi program sehingga lebih terstruktur, Mengurangi duplikasi kode dengan reusabilitas dan Mempermudah debugging karena setiap bagian memiliki fungsi spesifik.

Kekurangan: Overhead waktu eksekusi karena pemanggilan prosedur membutuhkan waktu tambahan dan terlalu banyak prosedur dapat membuat kode sulit dikelola jika tidak terorganisir dengan baik

II. GUIDED

Contoh Soal 1

```go
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"

//Prosedur untuk menghitung gaji karyawan
func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
    bonusLembur:= float64(jamLembur) * 50000
    totalGaji:= gajiPokok+bonusLembur

    fmt.Println("\n=== Slip Gaji ===")
    fmt.Println("Nama Karyawan :", nama)
    fmt.Printf("Gaji Pokok : Rp%.2f\n", gajiPokok)
    fmt.Printf("Bonus Lembur : Rp%.2f (%d jam x Rp50,000)\n",
bonusLembur, jamLembur)
    fmt.Printf("Total Gaji : Rp%.2f\n", totalGaji)
}

func main() {
    var nama string
    var gajiPokok float64
    var jamLembur int

    //Input dari pengguna
    fmt.Print("Masukkan Nama Karyawan: ")
    fmt.Scanln(&nama)

    fmt.Print("Masukkan Gaji Pokok: ")
    fmt.Scanln(&gajiPokok)

    fmt.Print("Masukkan Jumlah Jam Lembur: ")
    fmt.Scanln(&jamLembur)

    //Memanggil prosedur dengan data dari pengguna
    hitungGaji(nama, gajiPokok, jamLembur)
}
```

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODUL4> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODUL4\103112480280_Guided4\ContohSoal1.go"
Masukkan Nama Karyawan: Anggun
Masukkan Gaji Pokok: 30000000
Masukkan Jumlah Jam Lembur: 5

=== Slip Gaji ===
Nama Karyawan : Anggun
Gaji Pokok : Rp30000000.00
Bonus Lembur : Rp250000.00 (5 jam x Rp50,000)
Total Gaji : Rp30250000.00
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODUL4> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODUL4\103112480280_Guided4\ContohSoal1.go"
```

Deskripsi Program

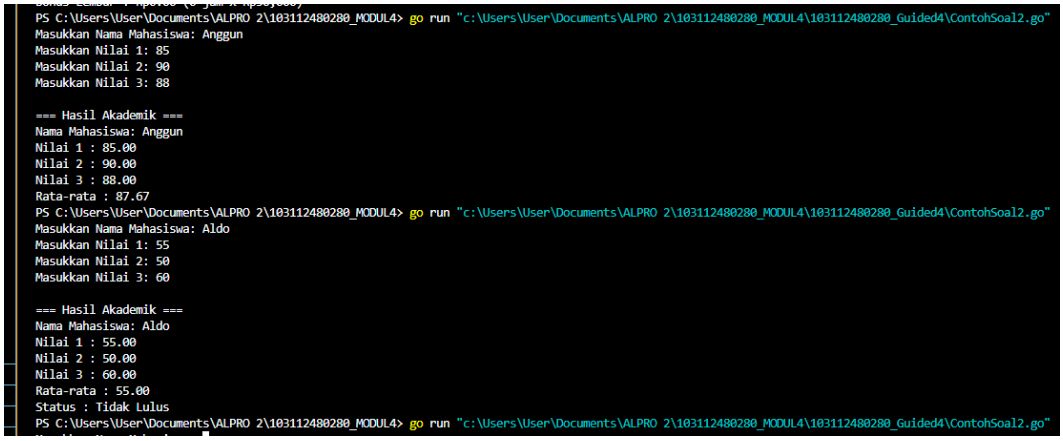
Program ini berfungsi untuk menghitung gaji karyawan berdasarkan gaji pokok dan jumlah jam lembur yang dikerjakan. Program ini meminta pengguna untuk memasukkan nama karyawan, gaji pokok, dan jumlah jam lembur. Setelah menerima input, program akan menghitung total gaji dengan menambahkan bonus lembur yang dihitung berdasarkan jumlah jam lembur yang dikerjakan. Bonus lembur ditetapkan sebesar Rp50.000 per jam. Hasil perhitungan gaji akan ditampilkan dalam format slip gaji yang mencakup nama karyawan, gaji pokok, bonus lembur, dan total gaji.

Contoh Soal 2

```go
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"

//Prosedur untuk menghitung rata-rata dan menentukan kelulusan
func hitungRataRata(nama string, nilai1, nilai2, nilai3 float64) {
    rataRata:=(nilai1+nilai2+nilai3) / 3
    status:= "Tidak Lulus"
    if rataRata >= 60 {
        status="Lulus"
    }

    fmt.Println("\n=== Hasil Akademik ===")
    fmt.Println("Nama Mahasiswa:", nama)
    fmt.Printf("Nilai 1 : %.2f\n", nilai1)
    fmt.Printf("Nilai 2 : %.2f\n", nilai2)
    fmt.Printf("Nilai 3 : %.2f\n", nilai3)
    fmt.Printf("Rata-rata : %.2f\n", rataRata)
    fmt.Println("Status :", status)
}

func main() {
    var nama string
    var nilai1, nilai2, nilai3 float64

    //Input dari pengguna
    fmt.Print("Masukkan Nama Mahasiswa: ")
    fmt.Scanln(&nama)

    fmt.Print("Masukkan Nilai 1: ")
    fmt.Scanln(&nilai1)

    fmt.Print("Masukkan Nilai 2: ")
```

```
    fmt.Scanln(&nilai2)

    fmt.Print("Masukkan Nilai 3: ")
    fmt.Scanln(&nilai3)

    // Memanggil prosedur dengan data dari pengguna
    hitungRataRata(nama, nilai1, nilai2, nilai3)
}
```

Output:



Deskripsi Program

Program ini berfungsi untuk menghitung rata-rata nilai akademik mahasiswa dan menentukan status kelulusan berdasarkan nilai yang dimasukkan. Program meminta pengguna untuk memasukkan nama mahasiswa dan tiga nilai yang diperoleh. Setelah menerima input, program akan menghitung rata-rata dari ketiga nilai tersebut. Jika rata-rata nilai mencapai 60 atau lebih, mahasiswa dinyatakan "Lulus"; jika tidak, dinyatakan "Tidak Lulus". Hasil perhitungan akan ditampilkan dalam format yang jelas.

III. UNGUIDED

Latihan Soal 1

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?

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 $\boldsymbol{a}$ 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!x2!) = 120/12 = 10 P(10,10) = 10!/0! = 3628800/1 = 3628800 C(10,10) = 10!/(10!x0!) = 10!/10! = 1
2.	8 0 2 0	56 28 1 1	

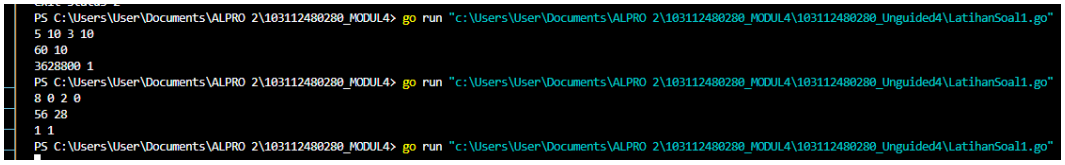
Selesaikan program tersebut dengan memanfaatkan subprogram yang diberikan berikut ini!

```
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}
```

```
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"
```

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

// Prosedur untuk menghitung permutasi
func permutation(n int, r int, hasil *int) {
    var faktorialN, faktorialNR int
    factorial(n, &faktorialN)
    factorial(n-r, &faktorialNR)
    *hasil = faktorialN / faktorialNR
}

// Prosedur untuk menghitung kombinasi
func combination(n int, r int, hasil *int) {
    var faktorialN, faktorialR, faktorialNR int
    factorial(n, &faktorialN)
    factorial(r, &faktorialR)
    factorial(n-r, &faktorialNR)
    *hasil = faktorialN / (faktorialR * faktorialNR)
}

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

    if a < c || b < d {
        fmt.Println("Input tidak valid! Pastikan a >= c dan b >=
d.")
        return
    }
    // Menghitung dan mencetak hasil permutasi dan kombinasi
    var hasilPermutasiAC, hasilKombinasiAC int
    var hasilPermutasiBD, hasilKombinasiBD int

    permutation(a, c, &hasilPermutasiAC)
    combination(a, c, &hasilKombinasiAC)

    permutation(b, d, &hasilPermutasiBD)
    combination(b, d, &hasilKombinasiBD)

    fmt.Println(hasilPermutasiAC, hasilKombinasiAC)
    fmt.Println(hasilPermutasiBD, hasilKombinasiBD)
}
```

Output:



Deskripsi Program

Program ini dirancang untuk menghitung nilai permutasi dan kombinasi berdasarkan empat bilangan bulat positif yang dimasukkan oleh pengguna: a, b, c, dan d. Dengan syarat bahwa a harus lebih besar atau sama dengan c, dan b harus lebih besar atau sama dengan d, program memvalidasi input sebelum melakukan perhitungan.

Program ini menggunakan tiga prosedur terpisah:

1. Prosedur factorial untuk menghitung faktorial dari bilangan bulat positif.
2. Prosedur permutation untuk menghitung nilai permutasi P(n,r) dengan memanfaatkan faktorial.
3. Prosedur combination untuk menghitung nilai kombinasi C(n,r) juga menggunakan faktorial.

Setelah perhitungan dilakukan, hasil permutasi dan kombinasi untuk kedua pasangan bilangan ditampilkan ke layar.

Latihan Soal 2

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 dala prosedur.

```
procedure 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.

Contoh

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.

```go
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"

// Prosedur untuk menghitung skor peserta
func hitungSkor(soal [8]int, totalSoal *int, totalWaktu *int) {
    *totalSoal = 0
    *totalWaktu = 0
    for _, waktu := range soal {
        if waktu <= 301 { // Jika waktu kurang dari atau sama dengan
301 menit
            *totalSoal++
            *totalWaktu += waktu
        }
    }
}

func main() {
    var nama1, nama2 string
    var soal1, soal2 [8]int
    var totalSoal1, totalWaktu1, totalSoal2, totalWaktu2 int

    // Input data peserta pertama
    fmt.Print("Masukkan nama peserta pertama: ")
    fmt.Scan(&nama1)
    fmt.Print("Masukkan waktu pengerjaan soal (8 angka):")
    for i := 0; i < 8; i++ {
        fmt.Scan(&soal1[i])
    }

    // Hitung skor peserta pertama
    hitungSkor(soal1, &totalSoal1, &totalWaktu1)

    // Input data peserta kedua
```

```
    fmt.Print("Masukkan nama peserta kedua: ")
    fmt.Scan(&nama2)
    fmt.Print("Masukkan waktu pengerjaan soal (8 angka):")
    for i := 0; i < 8; i++ {
        fmt.Scan(&soal2[i])
    }

    // Hitung skor peserta kedua
    hitungSkor(soal2, &totalSoal2, &totalWaktu2)

    // Menentukan pemenang
    if totalSoal1 > totalSoal2 || (totalSoal1 == totalSoal2 &&
totalWaktu1 < totalWaktu2) {
        fmt.Printf("%s %d %d\n", nama1, totalSoal1, totalWaktu1)
    } else {
        fmt.Printf("%s %d %d\n", nama2, totalSoal2, totalWaktu2)
    }
}
```

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODUL4> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODUL4\103112480280_Unguided4\LatihanSoal2.go"
Masukkan nama peserta pertama: Astuti
Masukkan waktu pengerjaan soal (8 angka):20 50 301 301 61 71 75 10
Masukkan nama peserta kedua: Bertha
Masukkan waktu pengerjaan soal (8 angka):25 47 301 26 50 60 65 21
Bertha 8 595
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODUL4> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODUL4\103112480280_Unguided4\LatihanSoal2.go"
Masukkan nama peserta pertama: Anggun
Masukkan waktu pengerjaan soal (8 angka):30 45 50 60 70 80 90 100
Masukkan nama peserta kedua: Aldo
Masukkan waktu pengerjaan soal (8 angka):25 50 301 302 310 290 280 275
Anggun 8 525
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODUL4> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODUL4\103112480280_Unguided4\LatihanSoal2.go"
```

Deskripsi Program

Program ini berfungsi untuk menentukan pemenang dalam kompetisi pemrograman dengan membandingkan jumlah soal yang diselesaikan dan total waktu yang digunakan oleh dua peserta. Setiap peserta diminta untuk memasukkan nama mereka dan waktu pengerjaan untuk delapan soal.

Program ini menggunakan prosedur hitungSkor, yang menghitung jumlah soal yang berhasil diselesaikan (dengan waktu kurang dari atau sama dengan 301 menit) dan total waktu yang dihabiskan untuk soal-soal tersebut.

Setelah menerima input dari kedua peserta, program memanggil prosedur untuk menghitung skor masing-masing. Pemenang ditentukan berdasarkan siapa yang menyelesaikan lebih banyak soal. Jika jumlah soal yang diselesaikan sama, pemenang ditentukan berdasarkan total waktu terendah.

Latihan Soal 3

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 cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

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

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

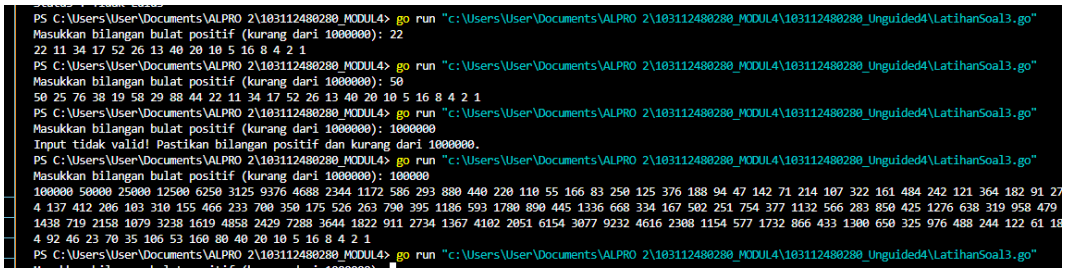
Keluaran terdiri dari satu baris saja. Setiap suku dari deret tersebut dicetak dalam baris yang dan dipisahkan oleh sebuah spasi.

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

```go
//Nama : Anggun Wahyu W. (103112480280)
package main

import "fmt"

// Prosedur untuk mencetak deret berdasarkan nilai awal
func cetakDeret(n int) {
    for n != 1 {
        fmt.Print(n, " ")
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Print(n) // Mencetak suku terakhir yaitu 1
}

func main() {
    var n int
```

```
    fmt.Print("Masukkan bilangan bulat positif (kurang dari
1000000): ")
    fmt.Scan(&n)

    if n <= 0 || n >= 1000000 {
        fmt.Println("Input tidak valid! Pastikan bilangan positif
dan kurang dari 1000000.")
        return
    }

    // Memanggil prosedur untuk mencetak deret
    cetakDeret(n)
}
```

Output:



Deskripsi Program

Program ini berfungsi untuk mencetak deret bilangan berdasarkan aturan yang ditentukan oleh Skiena dan Revilla. Deret dimulai dari sebuah bilangan bulat positif n, dan setiap suku berikutnya dihitung berdasarkan apakah bilangan tersebut genap atau ganjil. Jika genap, suku berikutnya adalah setengah dari bilangan tersebut (n / 2), sedangkan jika ganjil, suku berikutnya adalah tiga kali bilangan tersebut ditambah satu (3*n + 1). Deret berakhir ketika mencapai nilai 1.

Program menggunakan prosedur cetakDeret, yang menerima nilai awal n sebagai parameter dan mencetak setiap suku dari deret secara berurutan, dipisahkan oleh spasi. Input divalidasi untuk memastikan bahwa nilai yang dimasukkan adalah bilangan bulat positif kurang dari 1.000.000. Dengan struktur sederhana dan modular, program ini mencetak deret sesuai aturan hingga mencapai suku terakhir bernilai 1.

IV. KESIMPULAN

Prosedur adalah salah satu konsep penting dalam algoritma dan pemrograman yang digunakan untuk menyederhanakan kode dengan membagi tugas menjadi bagian-bagian kecil. Dalam laporan ini, prosedur diterapkan untuk menghitung skor peserta kompetisi berdasarkan waktu pengerjaan soal.

Program yang dibuat meminta input nama peserta dan waktu pengerjaan 8 soal, lalu menghitung jumlah soal yang dikerjakan dalam waktu ≤301 menit serta total waktu pengerjaan. Pemenang ditentukan berdasarkan jumlah soal yang dikerjakan, dan jika jumlahnya sama, berdasarkan total waktu yang lebih sedikit.

Melalui penerapan prosedur, program menjadi lebih modular, terstruktur, dan mudah dipahami.

V. REFERENSI

School of Computing. *Modul Praktikum 4 – Prosedur Algoritma dan Pemrograman 2 S1 Informatika*. 2025

Munir, Rinaldi. (2020). *Algoritma dan Pemrograman dalam Bahasa Go*. Bandung: Informatika.

Utami, Ema dan Sukrisno. (2019). *Konsep Dasar Pemrograman Go*. Yogyakarta: Graha Ilmu.