

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 6

MATERI



Oleh:

NAMA:Muhammad Fahruli Ma'ruf

NIM:103112400057

KELAS:12-IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Bahasa Pemrograman Go memiliki struktur program inti yang terdiri dari package main dan func main(), serta mendukung berbagai tipe data dasar seperti integer, float, boolean, char, dan string. Semua proses pengembangan Go dibantu oleh utilitas go (build, fmt, clean).

Dalam pemrograman Go, ada dua jenis subprogram utama: Fungsi yang selalu mengembalikan nilai dan digunakan untuk komputasi, serta Prosedur yang tidak mengembalikan nilai tetapi menjalankan serangkaian instruksi untuk menghasilkan efek langsung.

Selain itu, Go juga mendukung Rekursi, sebuah teknik penyelesaian masalah dengan memecah masalah menjadi sub-masalah identik melalui pemanggilan diri sendiri oleh subprogram, yang memerlukan base case sebagai kondisi berhenti dan recursive case untuk pemanggilan berulang.

Untuk organisasi data, Go menyediakan tipe bentukan seperti alias untuk penamaan ulang tipe data, dan struct untuk mengelompokkan data terkait. Go juga menawarkan struktur data koleksi yaitu array (dengan ukuran tetap) dan slice (array dinamis yang ukurannya dapat diubah menggunakan fungsi seperti len(), cap(), dan append()).

II. GUIDED

```
III. package main
IV.
V. import (
VI.     "fmt"
VII.     "time"
VIII. )
IX. // Struct untuk barang dalam struk belanja
X. type Item struct {
XI.     Name      string
XII.     Price     float64
XIII.    Quantity  int
XIV. }
XV. // Struct untuk struk belanja
XVI. type Receipt struct {
XVII.     StoreInfo string
XVIII.    Date      time.Time
XIX.    Items     []Item
XX.    TotalAmount float64
XXI. }
XXII. // Method untuk menghitung total harga semua item
XXIII. func (r *Receipt) CalculateTotal() {
XXIV.     var total float64
XXV.     for _, item := range r.Items {
XXVI.         total += item.Price * float64(item.Quantity)
XXVII.     }
XXVIII.    r.TotalAmount = total
XXIX. }
XXX. // Method untuk mencetak struk belanja
XXXI. func (r Receipt) PrintReceipt() {
XXXII.     fmt.Println("=====")
XXXIII.    fmt.Println(r.StoreInfo)
XXXIV.    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
XXXV.    fmt.Println("=====")
XXXVI.    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga",
XXXVII.    "Jumlah", "Total")
XXXVIII.    fmt.Println("-----")
XXXIX.    for _, item := range r.Items {
XL.        itemTotal := item.Price * float64(item.Quantity)
XLI.        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n",
XLII.        item.Name, item.Price, item.Quantity, itemTotal)
XLIII.    }
XLIV.    fmt.Println("=====")
```

```

XLIV.         fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:",
XLV.           r.TotalAmount)
XLVI.         fmt.Println("=====")
XLVII.        fmt.Println("Terima kasih telah berbelanja!")
XLVIII.       }
XLIX.       func main() {
L.           receipt := Receipt{
LI.             StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123,
LII.            Jakarta",
LIII.            Date:      time.Now(),
LIV.            Items: []Item{
LV.                {Name: "Beras", Price: 12000, Quantity: 5},
LVI.                {Name: "Gula", Price: 15000, Quantity: 2},
LVII.               {Name: "Minyak", Price: 20000, Quantity: 1},
LVIII.              {Name: "Telur", Price: 2000, Quantity: 10},
LIX.            },
LX.          }
LXI.          receipt.CalculateTotal()
           receipt.PrintReceipt()
           }

```

Penjelasan: Ketika program dijalankan, ia akan membuat sebuah struk belanja contoh dengan beberapa item. Program secara otomatis menghitung total biaya belanja berdasarkan item dan kuantitasnya, kemudian mencetak seluruh detail struk belanja tersebut ke layar konsol, menampilkan informasi toko, tanggal, daftar barang beserta harga dan jumlahnya, serta total yang harus dibayar

```

PS C:\Users\HP\OneDrive\modul6> go run "c:\Users\HP\OneDrive\modul6\contoh1\1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 12-06-2025 07:26
=====
Item          Harga      Jumlah  Total
-----
Beras         Rp12000.00    5      Rp60000.00
Gula          Rp15000.00    2      Rp30000.00
Minyak        Rp20000.00    1      Rp20000.00
Telur         Rp2000.00    10      Rp20000.00
=====
Total Belanja:                      Rp130000.00
=====
Terima kasih telah berbelanja!

```

```

contoh2 > go 2go > main
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      // Deklarasi dan inisialisasi array nilai mahasiswa
9      nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}
10
11      fmt.Println("Data Nilai Mahasiswa:")
12      fmt.Println("=====")
13
14      // Menampilkan nilai per mahasiswa
15      for i, nilai := range nilaiMahasiswa {
16          fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
17      }
18
19      // Menghitung rata-rata nilai
20      var total int
21      for _, nilai := range nilaiMahasiswa {
22          total += nilai
23      }
24      rataRata := float64(total) / float64(len(nilaiMahasiswa))
25
26      fmt.Println("=====")
27      fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)
28
29      // Mencari nilai tertinggi dan terendah
30      tertinggi := nilaiMahasiswa[0]
31      terendah := nilaiMahasiswa[0]
32
33      for _, nilai := range nilaiMahasiswa {
34          if nilai > tertinggi {
35              tertinggi = nilai
36          }
37          if nilai < terendah {
38              terendah = nilai
39          }
40      }
41
42      fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
43      fmt.Printf("Nilai terendah: %d\n", terendah)
44
45      // Contoh array 2 dimensi
46      fmt.Println("\nContoh Array 2 Dimensi:")
47      fmt.Println("=====")
48
49      // Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
50      nilaiUjian := [3][2]int{
51          {80, 85},
52          {90, 75},
53          {70, 95},
54      }
55
56      // Menampilkan nilai ujian per mahasiswa
57      fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
58      for i, nilai := range nilaiUjian {
59          fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1, nilai[0], nilai[1])
60      }
61  }

```

Penjelasan: Program Go ini adalah simulator sederhana pembuatan struk belanja. Ia menggunakan **struct** untuk mendefinisikan item dan struk, serta **method** untuk menghitung total belanja dan mencetak detail struk ke konsol.

```

Data Nilai Mahasiswa:
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
Mahasiswa 3: 78
Mahasiswa 4: 88
Mahasiswa 5: 95
=====
Rata-rata nilai: 87.20
Nilai tertinggi: 95
Nilai terendah: 78

Contoh Array 2 Dimensi:
=====
Nilai Ujian Mahasiswa (Matematika, Bahasa):
Mahasiswa 1: Matematika = 80, Bahasa = 85
Mahasiswa 2: Matematika = 90, Bahasa = 75
Mahasiswa 3: Matematika = 70, Bahasa = 95

```

LXII. UNGUIDED

```

LXIII. package main
LXIV.
LXV.     import (
LXVI.         "fmt"
LXVII.         "math"
LXVIII.     )
LXIX.
LXX.     type Titik struct {
LXXI.         x, y int
LXXII.     }
LXXIII.
LXXIV.     type Lingkaran struct {
LXXV.         pusat Titik
LXXVI.         r      int
LXXVII.     }
LXXVIII.
LXXIX.     func jarak(a, b Titik) float64 {
LXXX.         dx := float64(a.x - b.x)
LXXXI.         dy := float64(a.y - b.y)
LXXXII.         return math.Sqrt(dx*dx + dy*dy)
LXXXIII.     }
LXXXIV.
LXXXV.     func DL(l Lingkaran, t Titik) bool {
LXXXVI.         return jarak(l.pusat, t) <= float64(l.r)
LXXXVII.     }
LXXXVIII.
LXXXIX.     func main() {

```

```

XC.      fmt.Println("Program Pengecekan Titik dalam Lingkaran")
XCI.      fmt.Println("=====")
XCII.
XCIII.    var cx1, cy1, r1 int
XCIV.    var cx2, cy2, r2 int
XCV.      var x, y int
XCVI.      fmt.Println("\nLingkaran 1:")
XCVII.     fmt.Print("Pusat (x y): ")
XCVIII.    fmt.Scan(&cx1, &cy1)
XCIX.      fmt.Print("Jari-jari: ")
C.        fmt.Scan(&r1)
CI.        fmt.Println("\nLingkaran 2:")
CII.       fmt.Print("Pusat (x y): ")
CIII.      fmt.Scan(&cx2, &cy2)
CIV.       fmt.Print("Jari-jari: ")
CV.        fmt.Scan(&r2)
CVI.       fmt.Println("\nTitik yang akan dicek:")
CVII.      fmt.Print("Koordinat (x y): ")
CVIII.     fmt.Scan(&x, &y)
CIX.       if r1 <= 0 || r2 <= 0 {
CX.        fmt.Println("\nError: Jari-jari harus lebih besar dari
0!")
CXI.        return
CXII.      }
CXIII.     l1 := Lingkaran{Titik{cx1, cy1}, r1}
CXIV.     l2 := Lingkaran{Titik{cx2, cy2}, r2}
CXV.       titik := Titik{x, y}
CXVI.      dalam1 := DL(l1, titik)
CXVII.     dalam2 := DL(l2, titik)
CXVIII.    fmt.Println("\nHasil Pengecekan:")
CXIX.      fmt.Printf("Titik (%d, %d) ", x, y)
CXX.       if dalam1 && dalam2 {
CXXI.        fmt.Println("berada di dalam lingkaran 1 dan 2")
CXXII.      } else if dalam1 {
CXXIII.      fmt.Println("berada di dalam lingkaran 1")
CXXIV.      } else if dalam2 {
CXXV.        fmt.Println("berada di dalam lingkaran 2")
CXXVI.      } else {
CXXVII.      fmt.Println("berada di luar lingkaran 1 dan 2")
CXXVIII.    }
CXXIX.    }
CXXX.

```

Penjelasan: Kode Go ini adalah program yang berfungsi untuk **mengecek posisi sebuah titik relatif terhadap dua lingkaran**. Pengguna akan diminta untuk mendefinisikan dua lingkaran (koordinat pusat dan jari-jari) serta satu titik yang akan diperiksa. Program ini memungkinkan Anda mendefinisikan dua lingkaran dan satu titik. Setelah itu, ia akan menghitung jarak dari titik tersebut ke pusat masing-masing lingkaran, membandingkannya dengan jari-jari, dan kemudian memberitahu Anda apakah titik itu berada di dalam salah satu, kedua, atau tidak di kedua lingkaran yang Anda definisikan.


```

PS C:\Users\HP\OneDrive\modul6> go run "c:\Users\HP\OneDrive\modul6\soal1\1.go"
Program Pengecekan Titik dalam Lingkaran
=====

Lingkaran 1:
Pusat (x y): 1 1 5
Jari-jari:
Lingkaran 2:
Pusat (x y): 8 8 4
Jari-jari:
Titik yang akan dicek:
Koordinat (x y): 2 2

Hasil Pengecekan:
Titik (2, 2) berada di dalam lingkaran 1
PS C:\Users\HP\OneDrive\modul6> go run "c:\Users\HP\OneDrive\modul6\soal1\1.go"
Program Pengecekan Titik dalam Lingkaran
=====

Lingkaran 1:
Pusat (x y): 1 2 3
Jari-jari:
Lingkaran 2:
Pusat (x y): 4 5 6
Jari-jari:
Titik yang akan dicek:
Koordinat (x y): 7 8

Hasil Pengecekan:
Titik (7, 8) berada di dalam lingkaran 2
PS C:\Users\HP\OneDrive\modul6> go run "c:\Users\HP\OneDrive\modul6\soal1\1.go"
Program Pengecekan Titik dalam Lingkaran
=====

Lingkaran 1:
Pusat (x y): 5 10 15
Jari-jari:
Lingkaran 2:
Pusat (x y): -15 4 20
Jari-jari:
Titik yang akan dicek:
Koordinat (x y): 0 0

Hasil Pengecekan:
Titik (0, 0) berada di dalam lingkaran 1 dan 2
PS C:\Users\HP\OneDrive\modul6> go run "c:\Users\HP\OneDrive\modul6\soal1\1.go"
Program Pengecekan Titik dalam Lingkaran
=====

Lingkaran 1:
Pusat (x y): 1 1 5
Jari-jari:
Lingkaran 2:
Pusat (x y): 8 8 4
Jari-jari:
Titik yang akan dicek:
Koordinat (x y): 15 20

Hasil Pengecekan:
Titik (15, 20) berada di luar lingkaran 1 dan 2

```

```

package main

import (
    "fmt"
    "math"
)

func main() {
    fmt.Println("Program Operasi Array")
    fmt.Println("=====")
    angka := []int{10, 20, 30, 40, 50, 60, 70, 80, 90, 100}
    fmt.Println("Array awal:", angka)
    fmt.Println("\nElemen dengan indeks genap: ")
    for i := 1; i < len(angka); i += 2 {
        fmt.Printf("%d ", angka[i])
    }
    fmt.Println()
    fmt.Println("Elemen dengan indeks ganjil: ")
    for i := 0; i < len(angka); i += 2 {
        fmt.Printf("%d ", angka[i])
    }
    fmt.Println()
    kelipatan := 3
    fmt.Printf("\nElemen dengan indeks kelipatan %d: ", kelipatan)
    for i := 0; i < len(angka); i++ {
        if i%kelipatan == 0 {
            fmt.Printf("%d ", angka[i])
        }
    }
    fmt.Println()
    hapus := 4
    if hapus >= 0 && hapus < len(angka) {
        angka = append(angka[:hapus], angka[hapus+1:]...)
        fmt.Printf("\nArray setelah menghapus indeks %d: %v\n", hapus, angka)
    } else {
        fmt.Println("\nError: Indeks tidak valid!")
        return
    }
    total := 0
    for _, v := range angka {
        total += v
    }
    rata := float64(total) / float64(len(angka))
    fmt.Printf("\nRata-rata: %.2f\n", rata)
    var sum float64
    for _, v := range angka {
        sel := float64(v) - rata
        sum += sel * sel
    }
    stdev := math.Sqrt(sum / float64(len(angka)))
    fmt.Printf("Standar deviasi: %.2f\n", stdev)
    cari := 60
    frekuensi := 0
    for _, v := range angka {
        if v == cari {
            frekuensi++
        }
    }
    fmt.Printf("\nFrekuensi angka %d: %d kali\n", cari, frekuensi)
}

```

Penjelasan: Kode Go ini adalah program komprehensif yang mendemonstrasikan berbagai operasi dasar pada array (lebih tepatnya slice dalam Go) dan perhitungan statistik sederhana. Program ini dimulai dengan sebuah daftar angka. Ia akan menampilkan angka-angka pada posisi genap dan ganjil. Kemudian, ia akan meminta Anda untuk memasukkan sebuah angka kelipatan dan akan menampilkan angka-angka dari daftar yang merupakan kelipatan dari angka tersebut. Setelah itu, Anda bisa mencoba menghapus sebuah angka dari daftar berdasarkan posisinya. Terakhir, program akan menghitung dan menampilkan total, rata-rata, standar deviasi, dan berapa kali angka 60 muncul dalam daftar setelah semua operasi sebelumnya.

```
PS C:\Users\mr\OneDrive\modul10> go run C:\Users\mr\OneDrive\modul10\soal12\2.go
Program Operasi Array
=====
Array awal: [10 20 30 40 50 60 70 80 90 100]

Elemen dengan indeks genap: 20 40 60 80 100
Elemen dengan indeks ganjil: 10 30 50 70 90

Elemen dengan indeks kelipatan 3: 10 40 70 100

Array setelah menghapus indeks 4: [10 20 30 40 60 70 80 90 100]

Rata-rata: 55.56
Standar deviasi: 30.23

Frekuensi angka 60: 1 kali
```

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Program Pencatatan Pertandingan Sepak Bola")
7     fmt.Println("=====")
8     var klubA, klubB string
9     fmt.Print("\nMasukkan nama klub A: ")
10    fmt.Scan(&klubA)
11    fmt.Print("Masukkan nama klub B: ")
12    fmt.Scan(&klubB)
13    var hasil []string
14    pertandingan := 1
15    totalA := 0
16    totalB := 0
17    fmt.Println("\nMasukkan skor pertandingan (negatif untuk mengakhiri)")
18    for {
19        var skorA, skorB int
20        fmt.Printf("\nPertandingan %d (%s vs %s): ", pertandingan, klubA, klubB)
21        fmt.Scan(&skorA, &skorB)
22        if skorA < 0 || skorB < 0 {
23            break
24        }
25        if skorA > skorB {
26            hasil = append(hasil, klubA)
27            totalA++
28        } else if skorB > skorA {
29            hasil = append(hasil, klubB)
30            totalB++
31        } else {
32            hasil = append(hasil, "Seri")
33        }
34        pertandingan++
35    }
36    fmt.Println("\nRingkasan Pertandingan")
37    fmt.Println("=====")
38    fmt.Printf("%s vs %s\n", klubA, klubB)
39    fmt.Printf("Jumlah pertandingan: %d\n", pertandingan-1)
40    fmt.Printf("Kemenangan %s: %d\n", klubA, totalA)
41    fmt.Printf("Kemenangan %s: %d\n", klubB, totalB)
42    fmt.Printf("Hasil seri: %d\n", len(hasil)-totalA-totalB)
43    fmt.Println("\nHasil Pertandingan:")
44    for i, h := range hasil {
45        fmt.Printf("Pertandingan %d: %s\n", i+1, h)
46    }
47
48    fmt.Println("\nPertandingan selesai")
49 }

```

Penjelasan: kode tersebut menunjukkan berbagai fundamental pemrograman Go:

1. **Struktur & Organisasi Kode:** Menggunakan package main, func main(), struct untuk tipe data kustom, method untuk perilaku objek, serta membedakan antara **fungsi (mengembalikan nilai)** dan **prosedur (tidak mengembalikan nilai)** untuk modularitas.

2. **Manajemen Data:** Mengoperasikan **array (tetap)** dan **slice (dinamis)**, termasuk inisialisasi, akses elemen, pencarian, penghapusan, dan perhitungan statistik (total, rata-rata, standar deviasi, frekuensi).
3. **Algoritma:** Menerapkan konsep **rekursi** (dengan base case dan recursive case) dan **iterasi (perulangan)** untuk menyelesaikan masalah matematis (pangkat, faktorial, Fibonacci, deret Collatz) dan pola grafis (pola bintang).
4. **Interaksi Pengguna & Aplikasi Sederhana:** Mengambil input dari pengguna dan menggunakannya untuk simulasi aplikasi praktis seperti pencatatan pertandingan, struk belanja, kompetisi skor, pengecekan titik dalam lingkaran, serta perhitungan gaji atau status kelulusan.

```
PS C:\Users\HP\OneDrive\modul18> go run C:\Users\HP\OneDrive\modul18\main.go
Klub A : Mu
Klub B : Inter
Pertandingan 1 : 2 0
Format terminal 1 : 2 : 0
Pertandingan 2 : 1 2
Format terminal 2 : 1 : 2
Pertandingan 3 : 2 2
Format terminal 3 : 2 : 2
Pertandingan 4 : 0 1
Format terminal 4 : 0 : 1
Pertandingan 5 : 3 2
Format terminal 5 : 3 : 2
Pertandingan 6 : 1 0
Format terminal 6 : 1 : 0
Pertandingan 7 : 5 2
Format terminal 7 : 5 : 2
Pertandingan 8 : 2 3
Format terminal 8 : 2 : 3
Pertandingan 9 : -1 2

Hasil 1 : Mu
Hasil 2 : Inter
Hasil 3 : Mu (2 : 0)
Hasil 4 : Inter (1 : 2)
Hasil 5 : Draw (2 : 2)
Hasil 6 : Inter (0 : 1)
Hasil 7 : Mu (3 : 2)
Hasil 8 : Mu (1 : 0)
Hasil 9 : Mu (5 : 2)
Hasil 10 : Inter (2 : 3)
Pertandingan selesai
```

```

1 package main
2
3 import (
4     "fmt"
5     "unicode"
6 )
7
8 const MAKS = 127
9
10 type Tabel [MAKS]rune
11
12 func isiArray(tab *Tabel, n *int) {
13     var kar rune
14     *n = 0
15     fmt.Println("Masukkan teks (akhiri dengan '.'):")
16
17     for *n < MAKS {
18         fmt.Scanf("%c", &kar)
19         if kar == '.' {
20             break
21         }
22         if unicode.IsSpace(kar) {
23             tab[*n] = unicode.ToLower(kar)
24             *n++
25         }
26     }
27 }
28
29 func cetakArray(tab Tabel, n int) {
30     fmt.Println("Teks yang diinput: ")
31     for i := 0; i < n; i++ {
32         fmt.Printf("%c", tab[i])
33     }
34     fmt.Println()
35 }
36
37 func balikArray(tab *Tabel, n int) Tabel {
38     var hasil Tabel
39     j := 0
40     for i := n - 1; i >= 0; i-- {
41         hasil[j] = tab[i]
42         j++
43     }
44     return hasil
45 }
46
47 func palindrom(tab Tabel, n int) bool {
48     balik := balikArray(&tab, n)
49     for i := 0; i < n; i++ {
50         if tab[i] != balik[i] {
51             return false
52         }
53     }
54     return true
55 }
56
57 func main() {
58     fmt.Println("Program Pengecekan Palindrom")
59     fmt.Println("-----")
60
61     var tab Tabel
62     var n int
63     isiArray(&tab, &n)
64     if n == 0 {
65         fmt.Println("Error: Teks kosong!")
66         return
67     }
68     cetakArray(tab, n)
69     fmt.Println("\nHasil Pengecekan:\n")
70     if palindrom(tab, n) {
71         fmt.Println("Teks tersebut adalah palindrom")
72     } else {
73         fmt.Println("Teks tersebut bukan palindrom")
74     }
75     fmt.Println("\nTeks terbalik: ")
76     balik := balikArray(&tab, n)
77     for i := 0; i < n; i++ {
78         fmt.Printf("%c", balik[i])
79     }
80     fmt.Println()
81 }

```

Penjelasan: Kumpulan kode ini menunjukkan dasar-dasar pemrograman Go secara komprehensif. Program-program ini mencakup:

- **Struktur Kode Go:** Penggunaan package main dan func main() sebagai titik masuk program, serta penerapan modularitas melalui **fungsi** (mengembalikan nilai) dan **prosedur** (tidak mengembalikan nilai) untuk mengurangi kerumitan kode.
- **Manajemen Data:** Penanganan koleksi data seperti **array** (ukuran tetap) dan **slice** (ukuran dinamis). Operasi yang ditunjukkan meliputi inisialisasi, manipulasi elemen (penghapusan, pencarian), dan perhitungan statistik dasar (rata-rata, standar deviasi, frekuensi).
- **Algoritma:** Implementasi konsep **rekursi** (fungsi memanggil dirinya sendiri, dengan base case dan recursive case sebagai inti) dan **iterasi** (perulangan) untuk menyelesaikan masalah matematis (faktorial, pangkat, deret Fibonacci, deret Collatz) dan menghasilkan pola (pola bintang).
- **Aplikasi Praktis:** Kode-kode ini juga mensimulasikan skenario dunia nyata seperti pencatatan pertandingan olahraga, pengelolaan data mahasiswa, kompetisi skor, pengecekan palindrom, hingga perhitungan sederhana seperti gaji lembur atau status kelulusan.

```
PS C:\Users\HP\OneDrive\modul6> go run "c:\Users\HP\OneDrive\modul6\soal4\4.go"
Program Pengecekan Palindrom
=====
Masukkan teks (akhiri dengan '.'):
KATAK '.'
Teks yang diinput: katak'

Hasil Pengecekan:
Teks tersebut bukan palindrom

Teks terbalik: 'katak
PS C:\Users\HP\OneDrive\modul6> go run "c:\Users\HP\OneDrive\modul6\soal4\4.go"
Program Pengecekan Palindrom
=====
Masukkan teks (akhiri dengan '.'):
SENANG '.'
Teks yang diinput: senang'

Hasil Pengecekan:
Teks tersebut bukan palindrom

Teks terbalik: 'gnanes
```

CXXXI. KESIMPULAN

CXXXII. REFERENSI