LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 8

MATERI

Oleh:

NAMA:MUHAMMAD FAHRULI MA'RUF

NIM:103112400057

KELAS:12-IF-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

- Dasar Teori**Pencarian Sekuensial (Sequential Search):** Mencari data dengan memeriksa setiap elemen dari awal hingga ditemukan atau semua elemen sudah diperiksa. Cocok untuk data yang tidak terurut.
- **Pencarian Biner (Binary Search):** Mencari data dalam koleksi yang **sudah terurut** dengan membagi dua area pencarian pada setiap langkah, berfokus pada separuh yang relevan. Lebih efisien daripada pencarian sekuensial untuk data terurut.

(Dasar Teori Tipe Data dan Koleksi):

- **Tipe Bentukan (Custom Types):**
  - **Alias (type):** Memberi nama baru pada tipe data yang sudah ada untuk keringkasan dan kejelasan.
  - **Struct:** Mengelompokkan beberapa data (nilai atau tipe berbeda) yang saling berhubungan ke dalam satu kesatuan logis.
- **Koleksi Data:**
  - **Array:** Wadah data dengan ukuran **tetap** yang ditentukan di awal program.
  - **Slice (Array Dinamis):** Wadah data fleksibel yang ukurannya **dapat berubah** (dinamis) selama program berjalan, didukung oleh fungsi len (panjang), cap (kapasitas), dan append (menambah elemen).

(Dasar Teori Rekursif):

- **Rekursif:** Metode pemecahan masalah di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan sub-masalah yang identik. Ini adalah alternatif dari perulangan.
- **Komponen Utama:**
  - **Base Case:** Kondisi atau bagian yang menghentikan proses rekursif, mencegah perulangan tak terbatas. Ini adalah komponen terpenting.
  - **Recursive Case:** Kondisi di mana fungsi memanggil dirinya sendiri kembali untuk memproses sub-masalah.

(Kesimpulan Prosedur):

- **Prosedur:** Potongan instruksi program yang dikelompokkan untuk mengurangi kompleksitas kode yang besar. Prosedur menghasilkan efek langsung ketika dipanggil, tetapi **tidak mengembalikan nilai** secara eksplisit. Tujuannya adalah menyediakan titik referensi untuk tugas-tugas kecil yang dapat dipicu oleh programmer.

II. GUIDED

```go
package main

import (
    "fmt"
    "sort"
)

func sequentialSearch(arr[]float64, target float64)(int, int){
    iterations := 0
    for i, val := range arr{
        iterations++
        fmt.Printf("sequential stop %d: cek arr [%d]=%.f\n", iterations, i, val)
        if val == targ var iterations int
            return i, iterations
        }
    }
    return -1, iterations
}
func binarySEarch(arr []float64,target float64)(int, int){
    iterations := 0
    low := 0
    high := len (arr)-1
    for low <= high {
        iterations++
        mid := (low + high)/ 2
        fmt.Printf("binary stop %d: cek arr[%d]= %.f\n", iterations, mid, arr[mid])

        if arr[mid ]== target {
            return mid, iterations
        }else if target < arr[mid]{
            high = mid -1
        }else{
            low = mid -1
        }
    }
    return -1, iterations
}
 func main(){
    //array awal
    data:= []float64{2, 7, 9, 1, 5, 6, 18, 13, 25, 20}
    target := 13.0

    fmt.Println("sequential search (data tidak perlu urut):")
    idxseq, iterseq := sequentialSearch(data, target)
    if idxseq != -1{
        fmt.Printf("hasil : ditemukan di indeks %d dalam %d langkah\n\n",idxseq, iterseq)
    }else{
        fmt.Printf("hasil : tidak ditemukan setelah %d langkah\n",iterseq)
    }

    //Binary earch array diurutkan
    sort.Float64s(data)
    fmt.Println("Binary search (setelah data diurutkan):")
    fmt.Println("Data terurut:", data)

    idxBin, iterBin := binarySEarch(data, target)
    if idxBin!= -1 {
        fmt.Printf("Hasil : Ditemukan indeks %d dalam %d langkah\n", idxBin,iterBin)
    }else{
        fmt.Printf("Hasil : Tidak ditemukan Setelah %d langkah\n",iterBin)
    }
 }
```

Penjelasan: Program ini bertujuan untuk:

1. Mencari suatu nilai (target) dalam array **menggunakan Sequential Search**.
2. Mencari nilai yang sama **menggunakan Binary Search** setelah array diurutkan.
3. Menampilkan **jumlah langkah pencarian** dan hasilnya.

```
PS C:\Users\HP\OneDrive\modul8> go run "c:\Users\HP\OneDrive\modul8\contoh1\1.go"
sequential search (data tidak perlu urut):
sequential step 1: cek arr [0]=2
sequential step 2: cek arr [1]=7
sequential step 3: cek arr [2]=9
sequential step 4: cek arr [3]=1
sequential step 5: cek arr [4]=5
sequential step 6: cek arr [5]=6
sequential step 7: cek arr [6]=18
sequential step 8: cek arr [7]=13
hasil : ditemukan di indeks 7 dalam 8 langkah

Binary search (setelah data diurutkan):
Data terurut: [1 2 5 6 7 9 13 18 20 25]
binary stop 1: cek arr[4]= 7
binary stop 2: cek arr[6]= 13
Hasil : Ditemukan indeks 6 dalam 2 langkah
```

```
package main

import (
    "fmt"
    "sort"
)

type mahasiswa struct {
    nama, nim, kelas, jurusan string
    ipk                       float64
}

type arrMhs [2023]mahasiswa

// Sequential Search berdasarkan nama
func SeqSearch_3(T arrMhs, n int, X string) (int, int) {
    var found int = -1
    var j int = 0
    var iterasi int = 0

    for j < n && found == -1 {
```

```
        iterasi++
        if T[j].nama == X {
            found = j
        }
        j++
    }
    return found, iterasi
}

// Binary Search berdasarkan NIM (data harus sudah terurut berdasarkan
nim)
func BinarySearch_3(T arrMhs, n int, X string) (int, int) {
    var found int = -1
    var med int
    var kr int = 0
    var kn int = n - 1
    var iterasi int = 0

    for kr <= kn && found == -1 {
        iterasi++
        med = (kr + kn) / 2
        if X < T[med].nim {
            kn = med - 1
        } else if X > T[med].nim {
            kr = med + 1
        } else {
            found = med
        }
    }
    return found, iterasi
}

func main() {
    var data arrMhs
    n := 10

    // Mengisi data secara manual
    data = arrMhs{
        {nama: "Ari", nim: "2201", kelas: "A", jurusan: "Informatika", ipk:
3.4},
        {nama: "Budi", nim: "2203", kelas: "A", jurusan: "Informatika",
ipk: 3.6},
        {nama: "Cici", nim: "2202", kelas: "B", jurusan: "Sistem
Informasi", ipk: 3.5},
        {nama: "Dina", nim: "2205", kelas: "A", jurusan: "Informatika",
ipk: 3.3},
```

```
        {nama: "Eko", nim: "2204", kelas: "B", jurusan: "Sistem
Informasi", ipk: 3.7},
        {nama: "Fajar", nim: "2206", kelas: "C", jurusan: "Informatika",
ipk: 3.1},
        {nama: "Gita", nim: "2209", kelas: "C", jurusan: "Informatika",
ipk: 3.8},
        {nama: "Hana", nim: "2208", kelas: "B", jurusan: "Sistem
Informasi", ipk: 3.2},
        {nama: "Iwan", nim: "2207", kelas: "C", jurusan: "Informatika",
ipk: 3.0},
        {nama: "Joko", nim: "2210", kelas: "A", jurusan: "Informatika",
ipk: 3.9},
    }

    // Pencarian Sequential Search berdasarkan nama
    namaDicari := "Fajar"
    idxSeq, iterSeq := SeqSearch_3(data, n, namaDicari)

    fmt.Printf("Sequential Search - Cari nama '%s'\n", namaDicari)
    if idxSeq != -1 {
        fmt.Printf("Ditemukan di indeks: %d, Iterasi: %d\n", idxSeq,
iterSeq)
    } else {
        fmt.Printf("Tidak ditemukan, Iterasi: %d\n", iterSeq)
    }

    // Urutkan data berdasarkan NIM untuk binary search
    sort.Slice(data[:n], func(i, j int) bool {
        return data[i].nim < data[j].nim
    })

    // Pencarian Binary Search berdasarkan NIM
    nimDicari := "2206"
    idxBin, iterBin := BinarySearch_3(data, n, nimDicari)

    fmt.Printf("\nBinary Search - Cari NIM '%s'\n", nimDicari)
    if idxBin != -1 {
        fmt.Printf("Ditemukan di indeks: %d, Iterasi: %d\n", idxBin,
iterBin)
    } else {
        fmt.Printf("Tidak ditemukan, Iterasi: %d\n", iterBin)
    }
}
```

Penjelasan: program ini menunjukkan bagaimana dua metode pencarian yang berbeda—satu sederhana untuk data tidak terurut (Sequential Search) dan satu lagi lebih cepat untuk data terurut (Binary Search)—dapat digunakan untuk menemukan informasi dalam kumpulan data mahasiswa, sekaligus membandingkan jumlah langkah yang diperlukan oleh masing-masing metode.

```
PS C:\Users\HP\OneDrive\modul8> go run "c:\Users\HP\OneDrive\modul8\contoh2\2.go"
Sequential Search - Cari nama 'Fajar'
Ditemukan di indeks: 5, Iterasi: 6

Binary Search - Cari NIM '2206'
Ditemukan di indeks: 5, Iterasi: 3
```

III. UNGUIDED

```
IV.  package main
 V.
VI.  import "fmt"
VII.
III. func main() {
 IX.     var suara int
  X.     hitungSuara := make(map[int]int)
 XI.     totalSuaraMasuk := 0
XII.     totalSuaraSah := 0
III.     totalSuaraTidakSah := 0
XIV.
 XV.     fmt.Println("Program Penghitungan Suara
     Pemilihan")
XVI.     fmt.Println("================================")
VII.
III.     fmt.Println("\nMasukkan nomor calon (1-20)")
XIX.     fmt.Println("Masukkan 0 untuk mengakhiri input")
 XX.
XXI.     for {
XII.         fmt.Print("\nNomor calon: ")
III.         _, err := fmt.Scan(&suara)
XIV.         if err != nil {
XXV.             fmt.Println("Input tidak valid! Masukkan
     angka saja.")
XVI.             var clearInput string
VII.             fmt.Scanln(&clearInput) // Membersihkan
     buffer input
III.             continue
XIX.         }
XXX.
XXI.         if suara == 0 {
XII.             break
III.         }
XIV.
XXV.         totalSuaraMasuk++
XVI.
VII.         if suara >= 1 && suara <= 20 {
III.             hitungSuara[suara]++
XIX.             totalSuaraSah++
 XL.         } else {
XLI.             totalSuaraTidakSah++
LII.             fmt.Println("Nomor calon tidak valid! (1-
     20)")
```

```
III             }
LIV         }
XLV
LVI         fmt.Println("\nHasil Penghitungan Suara")
VII         fmt.Println("======================")
III         fmt.Printf("Total suara masuk: %d\n",
    totalSuaraMasuk)
LIX         fmt.Printf("Total suara sah: %d\n", totalSuaraSah)
 L          fmt.Printf("Total suara tidak sah: %d\n",
    totalSuaraTidakSah)
 LI
LII         ketua, wakil := 0, 0
III         max1, max2 := 0, 0
LIV
 LV         for calon, jumlah := range hitungSuara {
LVI             if jumlah > max1 {
VII                 max2 = max1
III                 wakil = ketua
LIX                 max1 = jumlah
 LX                 ketua = calon
LXI             } else if jumlah > max2 {
XII                 max2 = jumlah
III                 wakil = calon
XIV             }
LXV         }
XVI
VII         if totalSuaraSah > 0 {
III             fmt.Println("\nPerolehan Suara per Calon:")
XIX             for i := 1; i <= 20; i++ {
LXX                 if hitungSuara[i] > 0 {
XXI                     fmt.Printf("Calon %2d: %d suara\n", i,
    hitungSuara[i])
XII                 }
III             }
XIV
XXV             fmt.Println("\nHasil Akhir:")
XVI             fmt.Printf("Ketua RT terpilih: Calon %d (%d
    suara)\n", ketua, max1)
VII             fmt.Printf("Wakil Ketua terpilih: Calon %d (%d
    suara)\n", wakil, max2)
III         } else {
XIX             fmt.Println("\nTidak ada suara sah yang
    masuk")
XXX         }
XXI     }
```

Penjelasan:Program ini menghitung jumlah suara masuk dan suara sah sederhana,lalu menentukan pemenang berdasarkan suara terbanyak

```
PS C:\Users\HP\OneDrive\modul8> go run "c:\Users\HP\OneDrive\modul8\soal1\1.go"
Program Penghitungan Suara Pemilihan
================================

Masukkan nomor calon (1-20)
Masukkan 0 untuk mengakhiri input

Nomor calon: 7

Nomor calon: 19

Nomor calon: 3

Nomor calon: 2

Nomor calon: 78
Nomor calon tidak valid! (1-20)

Nomor calon: 3

Nomor calon: 1

Nomor calon: -3
Nomor calon tidak valid! (1-20)

Nomor calon: 18

Nomor calon: 19

Nomor calon: 0

Hasil Penghitungan Suara
======================
Total suara masuk: 10
Total suara sah: 8
Total suara tidak sah: 2

Perolehan Suara per Calon:
Calon  1: 1 suara
Calon  2: 1 suara
Calon  3: 2 suara
Calon  7: 1 suara
Calon 18: 1 suara
Calon 19: 2 suara

Hasil Akhir:
Ketua RT terpilih: Calon 19 (2 suara)
Wakil Ketua terpilih: Calon 3 (2 suara)
```

```
package main

import "fmt"

func main() {
```

```
    fmt.Println("Program Penghitungan Suara
Pemilihan\n==================================")
    fmt.Println("Masukkan nomor calon (1-20) atau 0 untuk mengakhiri
input.")

    var suara int
    hitungSuara := [21]int{}
    totalSuaraMasuk := 0
    totalSuaraSah := 0

    for {
        fmt.Print("\nNomor calon: ")
        _, err := fmt.Scan(&suara)
        if err != nil {
            fmt.Println("Input tidak valid! Masukkan angka saja.")
            var dummy string
            fmt.Scanln(&dummy)
            continue
        }

        if suara == 0 {
            break
        }

        totalSuaraMasuk++

        if suara >= 1 && suara <= 20 {
            hitungSuara[suara]++
            totalSuaraSah++
        } else {
            fmt.Println("Nomor calon tidak valid! (1-20)")
        }
    }

    fmt.Println("\nHasil Penghitungan
Suara\n======================")
    fmt.Printf("Total suara masuk: %d\n", totalSuaraMasuk)
    fmt.Printf("Total suara sah: %d\n", totalSuaraSah)
    fmt.Printf("Total suara tidak sah: %d\n", totalSuaraMasuk-
totalSuaraSah)

    ketua, wakil := 0, 0
    max1, max2 := 0, 0

    for i := 1; i <= 20; i++ {
        if hitungSuara[i] > max1 {
```

```
            max2 = max1
            wakil = ketua
            max1 = hitungSuara[i]
            ketua = i
        } else if hitungSuara[i] > max2 {
            max2 = hitungSuara[i]
            wakil = i
        }
    }

    if totalSuaraSah > 0 {
        fmt.Println("\nPerolehan Suara per Calon:")
        for i := 1; i <= 20; i++ {
            if hitungSuara[i] > 0 {
                fmt.Printf("Calon %2d: %d suara\n", i, hitungSuara[i])
            }
        }

        fmt.Println("\nHasil Akhir:")
        fmt.Printf("Ketua RT terpilih: Calon %d (%d suara)\n", ketua,
max1)
        fmt.Printf("Wakil Ketua terpilih: Calon %d (%d suara)\n", wakil,
max2)
    } else {
        fmt.Println("\nTidak ada suara sah yang masuk")
    }
}
```

Penjelasan: Program ini menghitung jumlah suara masuk dan suara sah sederhana,lalu menentukan pemenang berdasarkan suara terbanyak

```
PS C:\Users\HP\OneDrive\modul8> go run "c:\Users\HP\OneDrive\modul8\soal1\1.go"
Program Penghitungan Suara Pemilihan
================================

Masukkan nomor calon (1-20)
Masukkan 0 untuk mengakhiri input

Nomor calon: 7

Nomor calon: 19

Nomor calon: 3

Nomor calon: 2

Nomor calon: 78
Nomor calon tidak valid! (1-20)

Nomor calon: 3

Nomor calon: 1

Nomor calon: -3
Nomor calon tidak valid! (1-20)

Nomor calon: 18

Nomor calon: 19

Nomor calon: 0

Hasil Penghitungan Suara
=====================
Total suara masuk: 10
Total suara sah: 8
Total suara tidak sah: 2

Perolehan Suara per Calon:
Calon  1: 1 suara
Calon  2: 1 suara
Calon  3: 2 suara
Calon  7: 1 suara
Calon 18: 1 suara
Calon 19: 2 suara

Hasil Akhir:
Ketua RT terpilih: Calon 19 (2 suara)
Wakil Ketua terpilih: Calon 3 (2 suara)
```

```
package main

import "fmt"

func main() {
    fmt.Println("Program Pencarian Elemen")
    var n, k int
    _, err := fmt.Scan(&n, &k)
```

```
    if err != nil || n <= 0 {
        fmt.Println("Input tidak valid.")
        return
    }

    data := make([]int, n)

    for i := 0; i < n; i++ {
        _, err := fmt.Scan(&data[i])
        if err != nil {
            fmt.Println("Input elemen tidak valid.")
            return
        }
    }

    pos := -1
    for i := 0; i < n; i++ {
        if data[i] == k {
            pos = i
            break
        }
    }

    if pos != -1 {
        fmt.Printf("%d\n", pos)
    } else {
        fmt.Println("TIDAK ADA")
    }
}
```

Penjelasan:Program di atas bertujuan unutk mencari posisi sebuah bilangan tertentu dalam sebuah daftar bilangan yang telah di urut secara menurun

```
PS C:\Users\HP\OneDrive\modul8> go run "c:\Users\HP\OneDrive\modul8\soa
l3\3.go"
Program Pencarian Elemen
12 534
1 3 8 16 32 123 323 323 534 543 823 999
8
PS C:\Users\HP\OneDrive\modul8> go run "c:\Users\HP\OneDrive\modul8\soa
l3\3.go"
Program Pencarian Elemen
12 535
1 3 8 16 32 323 323 534 543 823 999
TIDAK ADA
```

LXXXII. KESIMPULAN

LXXXIII. REFERENSI