LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 8
PENCARIAN NILAI ACAK PADA HIMPUNAN DATA

DISUSUN OLEH:

ANASTASIA ADINDA NARENDRA INDRIANTO

103112400085

S1 IF-12-01

DOSEN:

Dimas Fanny Hebrasianto Permadi, S.ST., M.KOM

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

1. Pengertian Bahasa Pemorgraman Golang

Golang adalah bahasa pemrograman yang memiliki sejumlah kelebihan yang tidak dimiliki bahasa pemrograman yang lainnya. Hadirnya bahasa pemrograman Go Language (Golang) semakin dirasakan oleh para pengembang. Tidak heran jika banyak orang yang mulai belajar bahasa pemrograman yang satu ini.
Golang merupakan bahasa pemrograman yang dibuat Google dan tujuannya untuk menyempurnakan bahasa pemrograman yang ada, seperti C, Phyton dan yang lainnya. Golang bisa jadi pilihan yang tepat saat membuat aplikasi baru.

2. Pengertian Input dan Output

i. *Input* atau masukan adalah data yang diberikan ke dalam program. Masukan ini bisa berasal dari berbagai sumber, seperti pengguna melalui keyboard, mouse, atau suara, dan juga bisa berasal dari sensor atau perangkat lain yang terhubung ke komputer. Dalam pemrograman, input diperlakukan sebagai bahan baku yang akan diproses oleh program.

ii. *Output* atau keluaran adalah hasil yang diproduksi oleh program setelah mengolah input. Output bisa berbentuk tampilan pada layar, cetakan pada printer, suara, ataupun penyimpanan data ke file. Inti dari program yang kita tulis sebenarnya adalah menghasilkan output yang bermakna dari input yang diberikan.

3. Pengertian Tipe Data

Tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu.Secara sederhana, pengertian tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu. Contoh paling sederhana dari tipe data adalah tipe data **integer** yang digunakan untuk menyimpan angka bulat atau tipe data **string** yang digunakan untuk menyimpan rangkaian karakter. Penggunaan tipe data yang benar dalam suatu program memastikan bahwa data diolah secara tepat dan mengurangi risiko kesalahan atau bug dalam program tersebut.

4. Fungsi Tipe Data

i. **Menentukan Jenis Nilai**: Tipe data memberi tahu program jenis nilai yang akan disimpan dalam variabel. Misalnya, jika kita ingin menyimpan angka, kita menggunakan tipe data integer atau float, sedangkan untuk menyimpan teks, kita menggunakan tipe data string. Ini membantu program memahami bagaimana cara menangani data tersebut.

ii. **Efisiensi Penggunaan Memori**: Setiap tipe data memerlukan jumlah memori yang berbeda. Jika kita memilih tipe data yang tepat, program bisa menggunakan memori lebih efisien. Misalnya, menggunakan tipe data yang lebih kecil untuk angka yang tidak terlalu besar akan menghemat ruang di memori.

iii. **Menjamin Konsistensi Data**: Dengan menentukan tipe data, kita memastikan bahwa variabel hanya bisa menyimpan jenis nilai yang sesuai. Misalnya, jika kita mendeklarasikan variabel sebagai tipe integer, program tidak akan bisa memasukkan teks atau jenis data lainnya ke dalamnya. Ini membantu menjaga agar data tetap konsisten dan sesuai dengan yang diharapkan.

iv. **Memudahkan Operasi pada Data**: Tipe data juga menentukan operasi apa saja yang bisa dilakukan pada data. Misalnya, kita bisa melakukan perhitungan

matematika pada angka, tetapi kita tidak bisa melakukan hal yang sama pada teks. Jadi, dengan menentukan tipe data yang tepat, kita bisa melakukan operasi yang sesuai dengan jenis data yang kita miliki.

5. **Pengertian If-Else dan Fungsinya**

   If-else adalah struktur percabangan dalam pemrograman yang digunakan untuk mengeksekusi kode berdasarkan suatu kondisi. Jika kondisi dalam if bernilai true, maka blok kode di dalamnya akan dijalankan. Jika tidak, program akan memeriksa kondisi dalam else if sebagai alternatif. Jika semua kondisi false, maka else akan dijalankan sebagai pilihan terakhir. Dengan menggunakan if-else, program dapat mengambil keputusan dan menjalankan instruksi yang sesuai berdasarkan kondisi yang diberikan.

   If-else berfungsi untuk mengontrol alur program dengan mengeksekusi kode berdasarkan suatu kondisi. If digunakan untuk memeriksa apakah suatu kondisi bernilai true, jika ya, maka blok kode di dalamnya akan dijalankan. Jika tidak, program dapat menggunakan else-if untuk mengevaluasi beberapa kondisi tambahan secara berurutan. Jika semua kondisi false, maka else akan dieksekusi sebagai pilihan terakhir. Selain itu, terdapat if-else bersarang, yang memungkinkan pengecekan kondisi di dalam kondisi lain untuk menangani keputusan yang lebih kompleks.

6. **Pengertian While Loop dan Fungsinya**

   While loop adalah metode perulangan yang mengeksekusi blok kode selama kondisi yang diberikan bernilai true dan akan berhenti ketika kondisi berubah menjadi false. Perulangan ini sangat berguna dalam kasus di mana jumlah iterasi belum diketahui secara pasti, seperti membaca input hingga valid atau menjalankan suatu proses hingga syarat tertentu terpenuhi. Fungsinya adalah;

   i. **Menjalankan Perulangan Berdasarkan Kondisi** – While loop memastikan suatu proses terus berjalan selama kondisi bernilai **true**.
   ii. **Mengatasi Iterasi yang Tidak Diketahui Jumlahnya** – Digunakan ketika jumlah perulangan tidak bisa ditentukan sejak awal, seperti menunggu input yang valid.
   iii. **Mengoptimalkan Kontrol Program** – Membantu mengelola eksekusi kode agar hanya berjalan saat kondisi tertentu terpenuhi, meningkatkan efisiensi program.

7. **Pengertian Fungsi dan Manfaatnya**

   Fungsi merupakan rangkaian instruksi yang menghasilkan suatu nilai dengan memetakkan input ke output tertentu. Dalam pemrograman, suatu subprogram dikategorikan sebagai fungsi apabila memiliki deklarasi tipe nilai yang dikembalikan dan menggunakan kata kunci return dalam tubuhnya. Fungsi digunakan dalam berbagai situasi, seperti menetapkan nilai ke suatu variabel melalui assignment, menjadi bagian dari suatu ekspresi, atau digunakan sebagai argumen dalam subprogram lain. Karena fungsi selalu menghasilkan nilai, penamaannya sebaiknya bersifat deskriptif dan mencerminkan hasil yang diberikan, seperti *median*, *rerata*, *nilaiTerbesar*, atau *selesai*.

   Function memungkinkan programmer membagi kode menjadi segmen-segmen kecil yang lebih terkelola, masing-masing melakukan bagian tertentu dari tugas yang lebih besar. Tidak hanya membantu dalam mengorganisasi kode secara lebih efisien, hal ini juga memudahkan pemeliharaan dan pengujian kode.

8. **Pengertian Prosedur dan Manfaatnya**
   Prosedur adalah kumpulan instruksi yang dikemas menjadi satu kesatuan untuk menyederhanakan kode dalam program besar. Prosedur tidak mengembalikan nilai karena tidak memiliki deklarasi tipe nilai kembalian dan tidak menggunakan kata kunci *return*. Ketika dipanggil, prosedur langsung memberikan efek pada program utama, seperti halnya instruksi dasar atau fungsi bawaan (*built-in*). Nama prosedur sebaiknya menggunakan kata kerja atau kata yang menggambarkan proses, seperti *cetak*, *hitungRerata*, atau *cariNilai*. Hal ini bertujuan agar lebih mudah dipahami dan sesuai dengan fungsinya dalam program. Dengan menggunakan prosedur, kode program menjadi lebih terstruktur, mudah dibaca, dan dikelola.
   Manfaat utama prosedur adalah meningkatkan keterbacaan dan keteraturan kode, sehingga lebih mudah dikelola dan diperbaiki. Dengan memecah program menjadi bagian-bagian kecil, prosedur juga membantu dalam mengurangi pengulangan kode (code redundancy), meningkatkan efisiensi, serta mempermudah debugging dan pengembangan program. Hal ini membuat program lebih modular dan fleksibel untuk diperbarui di masa mendatang.
9. **Pengertian Rekursif dan Fungsinya**
   Rekursi adalah teknik dalam pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah yang lebih kecil dari masalah utama. Pendekatan ini sering digunakan untuk menyelesaikan permasalahan yang dapat dipecah menjadi submasalah serupa, seperti pencarian dalam struktur data pohon atau perhitungan faktorial. Agar tidak berjalan tanpa henti, rekursi memerlukan kondisi dasar yang menentukan kapan fungsi harus berhenti memanggil dirinya sendiri. Fungsi rekursif memiliki keunggulan dalam menyederhanakan kode untuk masalah yang memiliki pola berulang, seperti algoritma Divide and Conquer atau pemrosesan data yang bersifat hierarkis. Namun, jika tidak dirancang dengan baik, rekursi dapat menyebabkan penggunaan memori yang berlebihan atau bahkan error karena stack overflow. Oleh karena itu, memahami kapan dan bagaimana menggunakan rekursi dengan efisien sangat penting dalam pemrograman.
10. **Pengertian Struck dan Array**
    Struck adalah kumpulan yang terdiri dari elemen-elemen dengan tipe data yang heterogen atau tidak sama. Struktur dapat dideklarasikan menggunakan kata kunci 'struct', dan menggunakan '.' (operator titik) untuk mengakses elemen-elemen tersebut. Array mengacu pada koleksi yang terdiri dari elemen-elemen yang homogen, yaitu bertipe data yang sama. Array dideklarasikan menggunakan '[]'. Array menggunakan subskrip/'[ ]' (kurung siku) untuk mengakses elemen-elemennya. Pada dasarnya, array adalah pointer yang menunjuk ke elemen pertama dari sebuah koleksi.
11. **Pencarian Nilai Max dan Min**
    Pencarian adalah proses umum dalam kehidupan sehari-hari maupun dalam pemrograman, seperti mencari nilai maksimum atau minimum dalam array. Algoritma pencarian ini dimulai dengan menjadikan elemen pertama sebagai nilai awal, lalu dibandingkan satu per satu dengan elemen lainnya dalam array. Jika ditemukan nilai yang lebih ekstrim, maka nilai tersebut akan diperbarui. Setelah semua elemen diperiksa, nilai maksimum atau minimum yang valid akan diperoleh sebagai hasil akhir.

12. Pengertian Sequential Search

Sequential Search merupakan metode pencarian data yang dilakukan dengan membandingkan elemen-elemen dalam sebuah larik secara berurutan, dimulai dari elemen pertama hingga elemen yang dicari ditemukan atau seluruh data selesai diperiksa. Teknik ini tergolong sederhana dan cocok digunakan untuk data yang belum terurut. Sequential Search bermanfaat dalam berbagai aplikasi, seperti pencarian data barang dalam basis data, serta dapat dimanfaatkan untuk menguji performa algoritma melalui pengukuran waktu proses pencarian.

II. GUIDED

1. Guide 1

Source Code:

```
package main

import (
    "fmt"
    "sort"
)

func sequentialSearch(arr []float64, target float64) (int, int) {
    iterations := 0
    for i, val := range arr {
        iterations++
        fmt.Printf("Squential Step %d: cek arr[%d] = %.1f\n", iterations, i, val)
        if val == target {
            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 Step %d cek arr[%d] = %.1f\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() {
    data := []float64{2, 7, 9, 1, 5, 6, 18, 13, 25, 20}
    target := 13.0

    fmt.Println("Squential 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\n", iterSeq)
    }

    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 di indeks %d dalam %d langkah\n", idxBin,
iterBin)
    } else {
        fmt.Printf("Hasil: Tidak ditemukan setelah %d langkah\n", iterBin)
    }
}
```

Output:

```
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL8\Guided\1.go"
Squential Search (data tidak perlu urut)
Squential Step 1: cek arr[0] = 2.0
Squential Step 2: cek arr[1] = 7.0
Squential Step 3: cek arr[2] = 9.0
Squential Step 4: cek arr[3] = 1.0
Squential Step 5: cek arr[4] = 5.0
Squential Step 6: cek arr[5] = 6.0
Squential Step 7: cek arr[6] = 18.0
Squential Step 8: cek arr[7] = 13.0
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 Step 1 cek arr[4] = 7.0
Binary Step 2 cek arr[7] = 18.0
Binary Step 3 cek arr[5] = 9.0
Binary Step 4 cek arr[6] = 13.0
Hasil: Ditemukan di indeks 6 dalam 4 langkah
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8>
```

Deskripsi Program:

Program ini membandingkan dua metode pencarian data, yaitu Sequential Search dan Binary Search, untuk mencari nilai dalam array bertipe float64. Sequential Search mencari secara berurutan tanpa perlu data terurut, sedangkan Binary Search membutuhkan data terurut terlebih dahulu dan mencari secara efisien dengan membagi ruang pencarian. Program menampilkan langkah-langkah pencarian, indeks tempat data ditemukan (jika ada), dan jumlah langkah yang diperlukan oleh masing-masing metode.

2. Guide 2

Source Code:

```
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
    temp := [10]mahasiswa{
        {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},
    }

    for i := 0; i < n; i++ {
        data[i] = temp[i]
    }

    // 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)
    }
}
```

Output:

```
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL8\Guided\2.go"
Sequential Search - Cari nama 'Fajar'
Ditemukan di indeks: 5, Iterasi: 6

Binary Search - Cari NIM '2206'
Ditemukan di indeks: 5, Iterasi: 3
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8>
```

Deskripsi Program:

Program ini melakukan pencarian data mahasiswa menggunakan dua metode pencarian: Sequential Search berdasarkan nama dan Binary Search berdasarkan NIM. Data mahasiswa disimpan dalam array bertipe struct yang berisi nama, NIM, kelas, jurusan, dan IPK. Sequential Search mencari data nama secara berurutan tanpa syarat urutan data, sementara Binary Search mencari berdasarkan NIM dengan syarat data sudah terurut. Program menampilkan hasil pencarian beserta jumlah iterasi yang dibutuhkan.

III. UNGUIDED

1. Unguided 1

Source Code:

```
// Anastasia Adinda Narendra Indianto
// 103112400085
package main

import (
    "fmt"
    "sort"
)

func main() {
    var anastasiaInput int
    var totalSuara, suaraSah int
    var suara [21]int     // indeks 1-20 untuk calon
    var calonAda [21]bool // menandai calon mana saja yang mendapat suara

    for {
        fmt.Scan(&anastasiaInput)

        if anastasiaInput == 0 {
            break
        }

        totalSuara++

        if anastasiaInput >= 1 && anastasiaInput <= 20 {
            suara[anastasiaInput]++
            suaraSah++
            calonAda[anastasiaInput] = true
        }
    }

    fmt.Printf("Suara masuk: %d\n", totalSuara)
    fmt.Printf("Suara sah: %d\n", suaraSah)

    var daftarCalon [20]int
    idx := 0
    for i := 1; i <= 20; i++ {
        if calonAda[i] {
            daftarCalon[idx] = i
            idx++
        }
    }

    sort.Ints(daftarCalon[:idx])
```

```
    for i := 0; i < idx; i++ {
        c := daftarCalon[i]
        fmt.Printf("%d: %d\n", c, suara[c])
    }
}
```

Output:

```
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL8\Unguided\1.go"
7 19 3 2 78 3 1 -3 18 19 0
Suara masuk: 10
Suara sah: 8
1: 1
2: 1
3: 2
7: 1
18: 1
19: 2
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8>
```

Deskripsi Program:

Program ini dibuat untuk menghitung hasil pemilihan ketua RT dari 20 calon yang tersedia. Data suara dimasukkan berupa angka, dan hanya angka antara 1 hingga 20 yang dianggap sah. Input dihentikan saat pengguna memasukkan angka 0. Program menghitung jumlah total suara yang masuk, jumlah suara sah, serta mencetak daftar calon yang memperoleh suara beserta jumlahnya dalam urutan menaik. Program ini juga telah divalidasi untuk mengabaikan suara tidak sah secara otomatis.

2. Unguided 2

Source Code:

```
// Anastasia Adinda Narendra Indrianto
// 103112400085
package main

import (
    "fmt"
    "sort"
)

func main() {
    var anastasiaInput int
    var totalSuara, suaraSah int
    var suara [21]int

    for {
        fmt.Scan(&anastasiaInput)
        if anastasiaInput == 0 {
```

```
            break
        }

        totalSuara++

        if anastasiaInput >= 1 && anastasiaInput <= 20 {
            suara[anastasiaInput]++
            suaraSah++
        }
    }

    fmt.Printf("Suara masuk: %d\n", totalSuara)
    fmt.Printf("Suara sah: %d\n", suaraSah)

    type calon struct {
        nomor, suara int
    }

    var calonSuara []calon
    for i := 1; i <= 20; i++ {
        if suara[i] > 0 {
            calonSuara = append(calonSuara, calon{nomor: i, suara: suara[i]})
        }
    }

    sort.Slice(calonSuara, func(i, j int) bool {
        if calonSuara[i].suara == calonSuara[j].suara {
            return calonSuara[i].nomor < calonSuara[j].nomor
        }
        return calonSuara[i].suara > calonSuara[j].suara
    })

    if len(calonSuara) > 1 {
        ketua := calonSuara[0]
        wakilKetua := calonSuara[1]
        fmt.Printf("Ketua RT: %d\n", ketua.nomor)
        fmt.Printf("Wakil ketua: %d\n", wakilKetua.nomor)
    } else if len(calonSuara) == 1 {
        ketua := calonSuara[0]
        fmt.Printf("Ketua RT: %d\n", ketua.nomor)
        fmt.Printf("Wakil ketua: Tidak ada calon lain\n")
    } else {
        fmt.Println("Tidak ada calon yang sah")
    }
}
```

Output:

```
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL8\Unguided\2.go"
7 19 3 2 78 3 1 -3 18 19 0
Suara masuk: 10
Suara sah: 8
Ketua RT: 3
Wakil ketua: 19
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8>
```

Deskripsi Program:

Program ini digunakan untuk menghitung suara dalam pemilihan ketua RT, di mana input berupa bilangan bulat yang mewakili nomor calon ketua RT. Program ini membaca suara hingga ditemukan angka 0 yang menandakan berakhirnya input. Program kemudian memvalidasi suara yang sah (nomor antara 1 hingga 20), menghitung total suara yang masuk, dan suara yang sah. Setelah itu, program mengurutkan calon ketua berdasarkan jumlah suara terbanyak dan memilih ketua serta wakil ketua RT. Jika terdapat lebih dari satu calon dengan jumlah suara terbanyak yang sama, maka pemilihan ketua dan wakil ketua akan didasarkan pada nomor calon yang lebih kecil.

3. Unguided 3

Source Code:

```
//Anastasia Adinda Narendra Indrianto
//103112400085
package main

import "fmt"

const NMAX = 1000000
var data [NMAX]int

func main() {
    var anastasiaN, k int
    fmt.Scan(&anastasiaN, &k)
    isiArray(anastasiaN)
    idx := posisi(anastasiaN, k)

    if idx == -1 {
        fmt.Println("TIDAK ADA")
    } else {
        fmt.Println(idx)
    }
}

func isiArray(anastasiaN int) {
    for i := 0; i < anastasiaN; i++ {
        fmt.Scan(&data[i])
```

```
    }
}

func posisi(anastasiaN, k int) int {
    low := 0
    high := anastasiaN - 1

    for low <= high {
        mid := (low + high) / 2
        if data[mid] == k {
            return mid
        } else if data[mid] < k {
            low = mid + 1
        } else {
            high = mid - 1
        }
    }
    return -1
}
```

Output:

```
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL8\Unguided\3.go"
12 534
1 3 8 16 32 123 323 323 534 543 823 999
8
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL8\Unguided\3.go"
12 535
1 3 8 16 32 123 323 323 534 543 823 999
TIDAK ADA
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL8>
```

Deskripsi Program:

Program di atas melakukan pencarian bilangan dalam sebuah array terurut menggunakan algoritma binary search. Program menerima dua input: anastasiaN, yang merupakan jumlah elemen dalam array, dan k, bilangan yang ingin dicari. Array yang berisi anastasiaN bilangan integer akan diinputkan secara berturut-turut. Program kemudian mencari posisi bilangan k dalam array dengan menggunakan binary search. Jika bilangan ditemukan, program akan mencetak posisi (indeks) bilangan tersebut. Jika tidak ditemukan, program akan mencetak "TIDAK ADA".

IV. KESIMPULAN

Program-program yang disajikan menunjukkan implementasi dua metode pencarian, yaitu *Sequential Search* dan *Binary Search*. *Sequential Search* memeriksa setiap elemen satu per satu, cocok untuk data yang tidak terurut. Sedangkan *Binary Search* lebih efisien untuk data terurut, membagi pencarian menjadi dua bagian. Program ini juga mengaplikasikan pencarian dalam konteks pemilihan ketua RT dan data mahasiswa, memperlihatkan bagaimana algoritma pencarian digunakan untuk berbagai jenis masalah. Pemilihan metode bergantung pada apakah data sudah terurut atau tidak.

V. REFRENSI

https://sko.dev/wiki/input-dan-output

https://codingstudio.id/blog/golang-adalah/

https://dif.telkomuniversity.ac.id/tipe-data-pemrograman/

https://rpubs.com/maulidyarahmah/829044

https://repository.unikom.ac.id/62967/1/Materi%20Pertemuan%204_Labview%201%2BWhile%20Loop%20%2B%20Shift%20Register.pdf

https://www.revou.co/kosakata/function

https://sko.dev/wiki/fungsi

https://drive.google.com/file/d/1tMnxOLAYoMqLB9cKyeJHwyFjmyVtKqMV/view?usp=sharing

https://janabadra.ac.id/2023/algoritma-rekursi/#:~:text=Rekursi%20adalah%20sebuah%20metode%20pengulangan,seperti%20definisi%20fungsi%20pada%20umumnya.

https://www.tutorialspoint.com/difference-between-array-and-structure

file:///C:/Users/Laptopku/Downloads/Modul%207%20-%20Praktikum%20Alpro%202.pdf

https://drive.google.com/file/d/1VcZfFeHB_x3nf8IxW7p31FI6V0Bfvdpn/view?usp=sharing